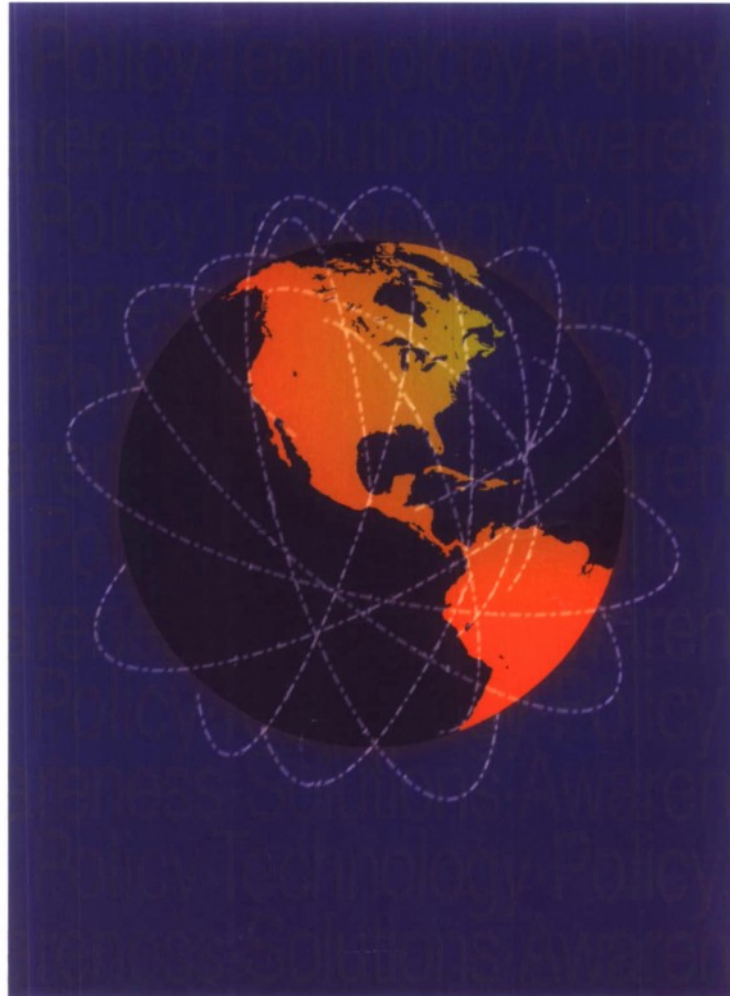


National Information Systems Security Conference

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY



NATIONAL COMPUTER SECURITY CENTER

20090327427

NATIONAL COMPUTER SECURITY CENTER

October 22-25, 1996
Baltimore Convention Center
Baltimore, MD

Volume I



DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

DTIC® has determined on 04/10/2009 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

☒ **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

☐ **© COPYRIGHTED;** U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

☐ **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors (fill in reason) (date of determination). Other requests for this document shall be referred to (insert controlling DoD office)

☐ **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only (fill in reason) (date of determination). Other requests shall be referred to (insert controlling DoD office).

☐ **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by (inserting controlling DoD office) (date of determination) or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

☐ **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25; (date of determination). DoD Controlling Office is (insert controlling DoD office).

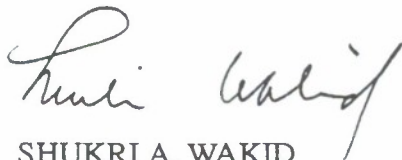
Welcome

The National Computer Security Center (NCSC) and the National Institute of Standards and Technology are pleased to welcome you to the Nineteenth National Information Systems Security Conference. We believe the conference will stimulate a productive information exchange and promote a greater understanding of today's information security issues and protection strategies.

The conference program addresses a wide range of interests from technical research and development projects to user-oriented management and administration topics. In today's ever more complex world where competitiveness demands swift, secure, value-added solutions, industry and government security professionals need to know how their vital information systems are threatened, what the vulnerabilities are, and how they can implement solutions. This Conference provides a unique international forum covering a wide variety of information systems security issues. Papers and panels in this multitrack program cover security issues related to: the Internet, electronic commerce, firewalls, information warfare, legal issues, computer crime, the World Wide Web, incident handling, cryptography, viruses, research and development, policies, vulnerabilities and threat, assurance, security engineering, and much more. As our technology increases, more enterprises are recognizing their need for computer security. The special sessions on electronic commerce and legal issues should be of particular interest to organizations that are starting to do business electronically.

The vendor exposition, sponsored by the Armed Forces Communications and Electronics Association (AFCEA) and held in parallel with this Conference, provides a forum for industry to showcase information systems security technology and provides hands-on demonstration of products and services that are potential solutions to many network and computer security problems.


We believe that the professional contacts you make at this conference, the presentations, and these Proceedings will offer you insights and ideas you can apply to your own security planning efforts. We encourage you to share the ideas and information you acquire this week with your peers, your management, and your customers. We also encourage you to share with us your successful security techniques as well as your thoughts and discussions about the problems you are experiencing and anticipate. It is through this exchange that we will continue to enhance the security of our information systems and networks and build a strong foundation to make security a credible value-added part of your enterprise such that security, policy, and technology truly are partners in your enterprise.



SHUKRI A. WAKID

Director

Computer Systems Laboratory



JOHN C. DAVIS

Director

National Computer Security Center

This Page
Intentionally
Left Blank

RISE OF THE MOBILE STATE:
ORGANIZED CRIME IN THE 21ST CENTURY

By

August Bequai, ESQ.
McLean, VA 22102

National Information Systems
Security Conference
October 22, 1996
Baltimore, MD

An associate of a New York Mafia family, is alleged to have orchestrated a multimillion dollar theft of microchips from a West Coast firm. A member of a European crime syndicate is said to have created fictitious accounts on the computers of a bank, and then used the funds to purchase securities. Members of an Asian crime family are said to have used the E-mail system of a multinational financial institution, to launder monies from their illegal operations.

Organized crime is a growth industry both within and outside the U.S. The fragmented global political environment has served to abet its growth. In the U.S. alone, organized crime is said to gross more than \$200 billion annually. No nation is immune from its tentacles. Security experts fear that the international crime syndicates are, increasingly, going high-tech. In large part, capitalizing on the implements of the IT revolution.

Asian, European, African, and Latin American crime syndicates are joining forces and pooling their resources; becoming a political and economic power in the global scene - a "mobile state", that rivals the multinational corporate giants in political and economic clout. Like the multinationals, the crime syndicates operate free of national restraints; guided by economic motives. In the process, they have harnessed the IT revolution.

Organized crime has learned to subvert IT so as to enhance its predatory practices; as well as augment its power and evade prosecution. Like the nomadic tribes of antiquity, who used the mobility of their fast steeds to prey on organized societies, these criminal mobile states are learning to implement EDI, the Internet, and other IT vehicles to their ends.

Why the Threat

Well into the 1980s, the international community, dismissed the threat of the global crime syndicates as the creation of Hollywood; while it made for good entertainment, it was not taken

seriously. Even the high-tech security establishment, fixated with hackers, focused little or no attention on the threat posed by the crime cartels. The IT security literature of the 1990s, replete with stories of cyber-crime and hackers, is noticeably devoid of any mention of organized crime; even a tangential one. The threat of syndicated crime in the IT environment, has been sublimated; nor have any efforts been made to study it.

The international crime syndicates have, historically, demonstrated an uncanny ability to employ the tools of technology in their arsenal. They have learned to adapt to their environment. The U.S. syndicates, and not the banks, made first extensive use of the wire services in the 1930s. The U.S. syndicates also employed, with success, the telephone, radio, air travel, and other technologies, to expand their operations over vast areas of North America. The growth of the U.S. Mafia in the 1930s can, in large part, be attributed to new technologies of that period. Its multibillion dollar gambling empire would not have been possible without the rise of telephonic communications. The Internet, should likewise, serve them well.

The crime syndicates have also demonstrated an ability to subvert both business and government. Blackmail, extortion and the threat of potential violence have been employed with noticeable success. In Italy, organized crime has even been able to topple governments; in Asia, the Triads and Yakuza helped their political allies gain political ascendancy. In Latin America, they have battled governments and left leaving revolutionary movements with success. They have demonstrated both the will and means to both survive and prevail.

But unfortunately, the international community has both neglected and underestimated the ability of the crime syndicates to employ the tools of IT in their illicit operations. While state-sponsored terrorism and the antics of religious zealots capture the daily headlines, the multibillion dollar EFT transactions of the drug cartel go unnoticed.

While modern terrorists constitute a growing problem, the ability and willingness of the crime cartels to terrorize and cause havoc, should not be dismissed. The Columbian syndicates have long since laid such doubts to rest.

But organized crime, even more so than the modern terrorists, is attuned to subtle vulnerabilities of the body politic of the nation-state. For example -

- (1) The crime syndicates have been known to extort monies from businesses and governments, in return for security. For example, the Asian syndicates were successful in keeping the extreme Left at bay, in return for political favors; in Italy, the Mafia decimated the Sicilian Communist party, in return for immunity from prosecution.

- (2) The syndicates have had little difficulty in coercing bankers to assist them in their money-laundering activities; or to tap into the multibillion dollar pension funds of labor unions.
- (3) The syndicates have been known to join forces with political radicals, when it meets their needs; as well as severing those alliances when their needs dictate otherwise. Asia and Latin are replete with examples of the drug cartels establishing alliances of convenience.
- (4) While the power base of the crime cartels is not based on geography, as is the case with the nation-state, they will exert control over defined territory when necessary. For example, the now defunct state of Herzeg Bosna served for a short period of time, as a haven for Balkan crime syndicates.

Exploiting the IT Revolution

While the IT revolution has amply demonstrated its worth, unfortunately, the environment in which it operates, is far from idyllic. The potential for criminal abuse is very real. Transnational crime syndicates operate with impunity in the current environment; the international organizations that were established to curtail their activities, have failed to do so. The syndicates not only prey on the user community; but they have also learned to employ the implements of IT to expand and enhance their control over their expanding illicit operations. EFT and related electronic payment systems, have dramatically facilitated the transborder movement of syndicate money.

Structure of the Syndicate

The very term syndicate or organized crime - these are frequently used interchangeably in the U.S., to denote organized criminal activity, as opposed to traditional street crime - evokes images of a handful of poorly educated individuals; from the lower strata of society, who meet secretly in dingy smoke-filled basements. Over the years, numerous efforts have been made in the U.S. and Europe to study and analyze the crime syndicates; the focus, however, has been on the European and U.S. Mafia groups. The Asian syndicates have largely escaped scrutiny. Hollywood continues to portray these groups as monoliths; dominated by chieftains of Mediterranean descent.

But organized crime is much more complex; as well as international in its operations. Crime syndicates permeate the societal fiber of every country. Some have their roots in Medieval History; evolving and adapting over the centuries. They go by different names - i.e., Yakuza, Triad, Camora, Mafia, Unione Corse, etc. - and exhibit diverse traits and modes of behavior. Some of them are historical rivals. But most of them share certain commonalities; among these -

- (1) Their basic structure and organization is largely feudal and highly decentralized; resembling the tribes and clans of the Medieval world, rather than the modern organizations that they prey on. Had they been monoliths, they would have proven easy to decapitate.
- (2) Their primary loyalty lies not with the nation-states from which they operate, but rather to the organization to which they belong; as well as its leadership.
- (3) Even the more sophisticated of the crime syndicates, idealizes the past; when civilization was less complex and simple. Post-industrial societies are viewed as decadent. The Yakuza, for example, look back fondly to the age of the Samurais; they view modern Japan with disdain.
- (4) While the syndicates pay lip-service to the idyllic past, they are driven by economic motives; selling their services to the highest bidder. For example, the Lebanese syndicates, while paying lip-service to Islam, sell their services to Muslims and Christians alike.
- (5) The syndicate families are bound together largely by kinship and blood ties. They often share a similar tradition and culture; as well as loyalty to the group. The nation-state and its laws, are merely tolerated.
- (6) The international syndicates are mobile in nature; with associates in many geographic areas. For example, the Triad syndicates have associates in Asia, North America and Europe.

While the criminal syndicates of the Medieval period operated, within confined geographic areas - the result of limitations imposed on them by the primitive technologies of their era - those of the IT society, operate globally. They make widespread use of IT to communicate with each other; as well as free themselves of the constraints of the nation-state. The IT revolution has given them mobility.

The Turning Point

Secret criminal societies have been with us since the dawn of civilization. They are the antithesis to organized government. The early twentieth century witnessed the rise and proliferation of criminal syndicates around the world; their expansion was abetted, in part, by the new technologies resulting from the industrial revolution. The urbanization of modern societies added fuel to their growth.

The turning point for the international syndicates came in the post-World War II period. Until then, the crime cartels had been fragmented, regional, and limited in their operations to

specified geographic areas. The post-World War II period witnessed the rise of new technologies and proliferation of new communication systems. Television became a household fixture. Armed with these technologies, the syndicates began to make their appearance on the global scene as powers to be reckoned with.

The new syndicate leadership, reared in the high-tech environment, turned its attention to international commerce. The syndicates embraced the world of high-technology; unfortunately, law enforcement failed to keep abreast. The modern syndicates must be viewed as a fusion of modern technology and a feudal organizational structure. This serves to make them dangerous to the post-industrial society; as well as impervious to its law enforcement apparatus.

Syndicates Embrace IT

IT lends itself to three key areas of syndicate activity: first, it makes the detection and prosecution of their illicit activities more difficult; secondly, it creates new targets of opportunity for them in the high-tech sector; and thirdly, it enhances their ability to coordinate and manage their global operations. With regard to the first, the failure of police agencies the world over to stay abreast of the IT revolution, has made the prosecution of the syndicates much more difficult.

Secondly, the IT revolution has opened new opportunities for the syndicates; i.e., computer/E-mail crimes, data thefts, computer sabotage, high-tech pornography, money laundering, and so forth. The third area, makes it possible for the syndicates to communicate by E-mail, EDI, and so on; it also serves to evidence their global mobility, and challenge the power of the nation-state.

High-Tech Crimes

IT has facilitated the commission of high-tech crimes by the syndicates. It can be employed to commit sophisticated wire frauds, commodity swindles, embezzlements, and other crimes. The multimillion dollar high-tech assisted swindles in the world of international finance, amply evidence the power of IT as a vehicle for the syndicates.

The syndicate have, over the years, been heavily involved in the financial frauds area. Syndicate controlled financial institutions, have been used in sophisticated high-tech frauds; as well as money laundering operations. The syndicate has also demonstrated an ability to employ IT in other endeavors. To cite a few examples -

- o Data thefts
- o Computer frauds and sabotage
- o EFT crimes
- o Bankruptcy frauds

- o Insurance scams
- o Securities swindles
- o Real estate scams
- o Industrial espionage
- o Theft of pension funds
- o Payoff and kickback schemes
- o Trafficking in stolen property

The use of IT in frauds against the government has also proved inviting to the syndicates; for example -

- o Diversion of government funds
- o Government contract frauds
- o Theft of confidential data
- o Sabotage of information systems
- o Tax frauds

The potential for misuse of IT by the syndicates is real and serious. The ability of the syndicates to prey on the post-industrial society has increased with the IT revolution. The latter has made it more difficult to secure the nation-state from syndicate attacks. The failure of the nation-state to develop the requisite tools to combat syndicate activities, has proven of help to the latter.

Going Cashless

The IT revolution has also prompted a revolution in the world of finance. Electronic payment systems now dominate international banking. Trillions of dollars are transferred by electronic means every hour. Efforts to secure these electronic systems from syndicate attack have fared ill.

Through the use of electronic banking systems, the syndicates can hide the billions of dollars that they collect from their drug trade and other illicit operations. IT has also provided the syndicates with necessary mobility to evade prosecution.

- (1) Extra-territorial activities by nation, aimed directly at the syndicates and their allies.
- (2) Mobile police forces, that can operate internationally.
- (3) IT safeguards to vend-off syndicated activities.

Aggressive steps need to also be taken by businesses to deter the illicit activities of the international syndicate. First and foremost, they need to enact security measures aimed at safeguarding their own IT systems. These should include -

- (1) Securing databases from unauthorized access, deletions, alterations and/or manipulation.

Combatting the Mobile State

Given their vast resources, the international crime syndicates pose a formidable challenge to the modern nation-state. Their mobility and transborder operations, hamper the traditional efforts of the nation-state to curtail their operations. Both international cooperation and programs are needed to deter and contain syndicate activities. These should include -

- o International mobile police forces that can traverse frontiers.
- o Treaties aimed at attacking the financial power bases of the syndicates.
- o Training for law enforcement agencies, in the detection, investigation, and prosecution of syndicate IT crimes.
- o Security measures for international networks, databases, EDI, E-mail, EFT, and related technologies.
- o Enhanced security awareness for both private and public officials.
- o Laws specifically directed at facilitating the prosecution of syndicate criminal activities.

Summary

The international crime syndicates are neither monoliths nor parochial in their operations. Asian syndicates have been known to work closely with their European and North American counterparts. While the various syndicates may differ in structure, organization, and motives, the IT revolution has accorded them new opportunities and enhanced mobility. They traverse the globe at-will; coordinating their efforts, in large part, through the vehicles of the IT revolution. Like the Mongols and other nomadic marauders of antiquity, they constitute mobile states. The IT revolution has given them a power base from whence they can threaten havoc to the nation-state; the latter must respond.

19th National Information Systems Security Conference

Co-Chairs

Stephen F. Barnett, *National Computer Security Center*
Tim Grance, *National Institute of Standards and Technology*

Program Directors

Ellen Flahavin, *National Institute of Standards and Technology*
Jack Holleran, *National Computer Security Center*

Program Committee

Edward Borodkin, *National Computer Security Center*
Christopher Bythewood, *National Computer Security Center*
Sally Meglarthary, *Estee Lauder*
Dr. Gary Smith, *Arca Systems*

Administration

Tammie Grice, *National Institute of Standards and Technology*
Mary Groh, *National Computer Security Center*
Kathy Kilmer, *National Institute of Standards and Technology*
C. A. O'Brien, *National Computer Security Center*
Melissa Petherbridge, *National Computer Security Center*
Phyllis Pierce, *National Computer Security Center*
Pat Purkey, *National Security Agency*
Sara Torrence, *National Institute of Standards and Technology*

Conference Referees

Dr. Marshall Abrams
Rowland Albert
James P. Anderson
Devolyn Arnold
James Arnold
Alfred Arsenault
Dr. D. Elliott Bell
Dr. Matt Bishop
Earl Boebert
Dr. Dennis Branstad
Dr. Martha Branstad
Dr. Blaine Burnham
Christopher Bythewood
Dr. William Caelli
Dr. John R. Campbell
Lisa Carnahan
Dr. Jon David
Dr. Dorothy E. Denning
Donna Dodson

The MITRE Corporation
National Security Agency
J. P. Anderson Company
National Security Agency
National Security Agency
National Security Agency
Mitretek Corporation
University of California, Davis
Sandia National Laboratory
Trusted Information Systems, Inc.
Trusted Information Systems, Inc.
National Security Agency
National Computer Security Center
Queensland University of Technology, Australia
National Security Agency
National Institute of Standards and Technology
The Fortress
Georgetown University
National Institute of Standards and Technology

Conference Referees *(continued)*

Karen Ferraiolo	<i>Arca Systems</i>
Ellen Flahavin	<i>National Institute of Standards and Technology</i>
Dan Gambel	<i>General Research Corporation</i>
Virgil Gibson, CISSP	<i>Computer Sciences Corporation</i>
Dennis Gilbert	<i>National Institute of Standards and Technology</i>
Barbara Guttman	<i>National Institute of Standards and Technology</i>
Dr. Grace Hammonds	<i>AGCS, Inc.</i>
Cindy Hash	<i>National Security Agency</i>
Ronda Henning	<i>Harris Corporation</i>
Dr. Harold Highland, FICS, FACM	<i>Computers & Security</i>
Jack Holleran	<i>National Computer Security Center</i>
Hillary H. Hosmer	<i>Data Security</i>
Carole Jordan	<i>Grumman Data Systems</i>
Steve Kougoures	<i>National Security Agency</i>
David Krehnke	<i>Lockheed Martin Energy Systems</i>
Helmut Kurth	<i>Industrieanlagen Betriebsghesellschaft mbH (IABG), Germany</i>
Carl Landwehr	<i>Naval Research Laboratory</i>
Robert Lau	<i>National Security Agency</i>
Dr. Theodore M. P. Lee	<i>Independent Consultant</i>
Special Agent John Lewis	<i>United States Secret Service</i>
Steven Lipner	<i>Trusted Information Systems, Inc.</i>
Joseph Lisi	<i>National Security Agency</i>
Teresa Lunt	<i>Defense Advanced Research Projects Agency</i>
Wayne Madsen	<i>Computer Sciences Corporation</i>
John McDermott	<i>J - K International Limited</i>
Dr. John McLean	<i>Naval Research Laboratory</i>
Sally Meglathery	<i>Estee Lauder</i>
William H. Murray	<i>Deloitte & Touche</i>
Ruth Nelson	<i>Information System Security</i>
Dr. Peter Neumann	<i>Stanford research Institute, International</i>
Dr. Charles Pfleeger	<i>Trusted Information Systems, Inc.</i>
W. Timothy Polk	<i>National Institute of Standards and Technology</i>
Marcus Ranum	<i>V-ONE</i>
Marvin Schaefer	<i>Arca Systems</i>
Dr. Gary Smith	<i>Arca Systems</i>
Dr. Eugene Spafford	<i>Coast Laboratory, Purdue University</i>
Julian Straw	<i>Syntegra, UK</i>
James Tippet	<i>National Security Agency</i>
Ken van Wyk	<i>Science Applications International Corporation</i>
John Wack	<i>National Institute of Standards and Technology</i>
Mark Wallace	<i>National Security Agency</i>
Howard Weiss	<i>SPARTA, Inc.</i>
Valerie Williams	<i>Data Sciences, UK</i>
Roy Wood	<i>National Security Agency</i>
Mark Woodcock	<i>National Security Agency</i>
Paul Woodie	<i>National Security Agency</i>
Thomas Zmudzinski	<i>Defense Information Systems Agency</i>

AWARDS CEREMONY

2:00 p.m. Thursday October 24

Baltimore Convention Center, Room 337-338

The National Institute of Standards and Technology (NIST) and the National Computer Security Center (NCSC) will honor those vendors who have successfully developed products meeting the standards of the respective organizations. Immediately following the ceremony, honored vendors will have the opportunity to display these products.

The NCSC recognizes vendors who contribute to the availability of trusted products and thus expand the range of solution from which customers may select to secure their data. The products are placed on the Evaluated Products List (EPL) following a successful evaluation against the Trusted Computer Systems Evaluation Criteria including its interpretations: Trusted Database Interpretation; Trusted Network Interpretation; and Trusted Subsystems Interpretation. Vendors who have completed the evaluation process will receive a formal certificate of completion from the Director, NCSC marking the addition to the EPL. Certificates will also be presented to those vendors that have placed a new release of a trusted product on the EPL by participation in the Ratings Maintenance Program (RAMP). Additionally, vendors will receive honorable mention for being in the final stages of an evaluation as evidenced by transition into the Formal Evaluation phase. The success of the Trusted Product Evaluation Program is made possible by the commitment of the vendor community.

The Computer Security Division at NIST provides validation services to test vendor implementations for conformance to security standards. NIST currently maintains validation services for three Federal Information Processing Standards (FIPS): FIPS 46-2, Data Encryption Standards (DES); FIPS 113, Computer Data Authentication; and FIPS 171, Key Management Using ANSI X9.17. During this award ceremony, NIST presents "Certificate of Appreciation" awards to those vendors who have successfully validated their implementation of these standards.

With the reaffirmation of the Data Encryption Standard as FIPS 46-2 in 1993, DES can now be implemented in software, as well as hardware and firmware. To successfully validate an implementation for conformance to FIPS 46-2, a vendor must run the Monte Carlo test as described in NBS (NIST) Special Publication 500-20. The Monte Carlo test consists of performing eight million encryptions and four million decryptions, with two encryptions and one decryption making a single test.

Vendors test their implementations of conformance to FIPS 113 and its American National Standards Institute (ANSI) counterpart, ANSI X9.9, Financial Institution Message Authentication (Wholesale). This is done using an electronic bulletin board system. Interactive validation requirements are specified in NBS (NIST) Special Publication 500-156, Message Authentication Code (MAC) Validation System: Requirements and Procedures. The test suite is composed of a series of challenges and responses in which the vendor is requested to either compute or verify a MAC on given data using a specified key which was randomly generated.

Conformance to FIPS 171 is also tested using an interactive electronic bulletin board testing suite. FIPS 171 adopts ANSI X9.17, Financial Institution Key Management (Wholesale). ANSI X9.17 is a key management standard for DES-based applications. The tests are defined in a document entitled NIST Key Management Validation System Point-to-Point (PTP) Requirements. The test suite consists of a sequence of scenarios in which protocol messages are exchanged under specified conditions.

We congratulate all who have earned these awards.

19th National Information Systems Security Conference

Welcome Letter	i
Keynote Speech: August Bequai, Esq.	iii
Conference Committee & Referees	x
Award Ceremony	xii
Table of Contents	xiii
Author Cross Reference	xxvii

Refereed Papers

Criteria & Assurance

Track A

E4 ITSEC Evaluation of PR/SM on ES/9000 Processors.....	1
Naomi Htoo-Mosher, Robert Nasser, Nevenko Zunic, <i>International Business Machines</i>	
Julian Straw, <i>SynTEGRA, UK</i>	
A High-Performance Hardware-Based High Assurance Trusted Windowing System.....	12
Jeremy Epstein, <i>Cordant, Inc.</i>	
WWW Technology in the Formal Evaluation of Trusted Systems	22
E.J. McCauley, <i>Silicon Graphics Computer Systems, Inc.</i>	
The Certification of the Interim Key Escrow System.....	26
Ellen Flahavin, Ray Snouffer, <i>National Institute of Standards and Technology</i>	
Configuration Management in Security related Software Engineering Processes	34
Klaus Keus, Thomas Gast, <i>Bundesamt für Sicherheit in der Informationstechnik, Germany</i>	
The Department of Defense Information Technology Security Certification and Accreditation Process (DITSCAP)	46
Jack Eller, <i>DISA</i>	
Mike Mastrococco, <i>Computer Security Consulting</i>	
Barry C. Stauffer, <i>CORBETT Technologies, Inc.</i>	
Trusted Process Classes	54
William L. Steffan, <i>Tracor Applied Science, Inc.</i>	
Jack D. Clow, <i>SenCom Corporation</i>	
Design Analysis in Evaluations Against the TCSEC C2 Criteria	67
Frank Belvin, Deborah Bodeau, Shaan Razvi, <i>The MITRE Corporation</i>	
System Security Engineering Capability Maturity Model and Evaluations: Partners within the Assurance Framework	76
Charles G. Menk III, <i>Department of Defense</i>	
Applying the TCSEC Guidelines to a Real-Time Embedded System Environment	89
Jim Alves-Foss, Deborah Frincke, Gene Saghi, <i>University of Idaho</i>	

Electronic Commerce

Track B

EDI Moves from the VAN to the Internet	98
Brian Bradford, <i>University of Maryland</i>	
An International Standard for the Labeling of Digital Products	109
Viktor E. Hampel, <i>Hampel Consulting</i>	

An International Standard for the Labeling of Digital Products	109
Viktor E. Hampel, <i>Hampel Consulting</i>	
The Business-LED Accreditor - OR...How to Take Risks and Survive.....	123
Michael E J Stubbings, <i>Government Communications Headquarters, UK</i>	
Integration of Digital Signatures into the European Business Register	131
Helmut Kurth, <i>Industrieanlagen Betriebsgesellschaft mbH, Germany</i>	
Industrial Espionage Today and Information Wars of Tomorrow	139
Paul M. Joyal, <i>INTEGER Inc.</i>	
B is for Business: Mandatory Security Criteria & the OECD Guidelines for Information Systems Security	152
Prof. William J. Caelli, <i>Queensland University of Technology, Australia</i>	
Marketing & Implementing Computer Security	163
Mark Wilson, <i>National Institute of Standards and Technology</i>	
Secure Internet Commerce - - Design and Implementation of the Security Architecture of Security First Network Bank, FSB.....	173
Nicolas Hammond, <i>NJH Security Consulting, Inc.</i>	

In Depth

Track C

Automatic Formal Analyses of Cryptographic Protocols	181
Stephen H. Brackin, <i>Arca Systems, Inc.</i>	
Surmounting the Effects of Lossy Compression on Steganography	194
Daniel L. Currie, III, <i>Fleet Information Warfare Center</i> Cynthia E. Irvine, <i>Naval Postgraduate School</i>	
Key Escrowing Systems and Limited One Way Functions.....	202
William T. Jennings, <i>Southern Methodist University & Raytheon E-Systems</i> James G. Dunham, <i>Southern Methodist University</i>	
The Keys to a Reliable Escrow Agreement	215
Richard Sheffield	

Internet

Track D

The Advanced Intelligent Network — A Security Opportunity	221
Thomas A. Casey, Jr., <i>GTE Laboratories, Inc.</i>	
Security Issues in Emerging High Speed Networks.....	233
Vijay Varadharajan, <i>University of Western Sydney, Australia</i> Panos Katsavos, <i>Hewlett Packard sponsored student, UK</i>	
A Case Study of Evaluating Security in an Open Systems Environment	250
Daniel L. Tobat, <i>TASC</i> Errol S. Weiss, <i>Science Applications International Corporation</i>	
Internet Firewalls Policy Development and Technology Choices	259
Leonard J. D'Alotto, <i>GTE Laboratories, Inc.</i>	

A Case for Avoiding Security-Enhanced HTTP Tools to Improve Security for Web-Based Applications.....	267
Bradley J. Wood, <i>Sandia National Laboratories</i>	
Applying the Eight Stage Risk Assessment Methodology to Firewalls.....	276
David L. Drake, Katherine L. Morse, <i>Science Applications International Corporation</i>	
Lessons Learned: An Examination of Cryptographic Security Services in a Federal Automated Information System.....	288
Jim Foti, Donna Dodson, Sharon Keller, <i>National Institute of Standards and Technology</i>	

Legal Perspectives

Track E

Intellectual Property Rights and Computer Software	296
Dawn E. Bowman, <i>University of Maryland</i>	
Case Study of Industrial Espionage Through Social Engineering	306
Ira S. Winkler, <i>National Computer Security Association</i>	
Legal Aspects of Ice-Pick Testing	313
Dr. Bruce C. Gabrielson, <i>Kaman Sciences Corp.</i>	

Management & Administration

Track F

Security Through Process Management.....	323
Jennifer L. Bayuk, <i>Price Waterhouse, LLP.</i>	
Malicious Data and Computer Security.....	334
W. Olin Sibert, <i>InterTrust Technologies Corporation</i>	
Security Issues for Telecommuting	342
Lisa J. Carnahan, Barbara Guttman, <i>National Institute of Standards and Technology</i>	

Research & Development

Track G

An Isolated Network for Research.....	349
Matt Bishop, L. Todd Heberlein, <i>University of California, Davis</i>	
GrIDS-A Graph-Based Intrusion Detection System for Large Networks.....	361
S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger, J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle, <i>University of California, Davis</i>	
Attack Class: Address Spoofing	371
L. Todd Heberlein, <i>Net Squared</i> Matt Bishop <i>University of California, Davis</i>	
Generic Model Interpretations: POSIX.1 and SQL	378
D. Elliott Bell, <i>Mitretek Systems</i>	
The Privilege Control Table Toolkit: An Implementation of the System Build Approach.....	389
Thomas R. Woodall, Roberta Gotfried, <i>Hughes Aircraft Company</i>	

Use of the Zachman Architecture for Security Engineering	398
Ronda Henning, <i>Harris Corporation</i>	
Developing Secure Objects.....	410
Deborah Frincke, <i>University of Idaho</i>	
Deriving Security Requirements for Applications on Trusted Systems.....	420
Raymond Spencer, <i>Secure Computing Corporation</i>	
Security Implications of the Choice of Distributed Database Management System	428
Model: Relational vs. Object-Oriented	
Stephen Coy, <i>University of Maryland</i>	
Management Model for the Federal Public Key Infrastructure.....	438
Noel A. Nazario, William E. Burr, W. Timothy Polk,	
<i>National Institute of Standards and Technology</i>	
Security Policies for the Federal Public Key Infrastructure	445
Noel A. Nazario, <i>National Institute of Standards and Technology</i>	
A Proposed Federal PKI using X.509 V3 Certificates	452
William E. Burr, Noel A. Nazario, W. Timothy Polk,	
<i>National Institute of Standards and Technology</i>	
A Security Flaw in the X.509 Standard.....	463
Santosh Chokhani, <i>CygnaCom Solutions, Inc.</i>	

Solutions

Track H

Computer Virus Response Using Autonomous Agent Technology.....	471
Christine M. Trently, <i>Mitretek Systems</i>	
Security Across the Curriculum: Using Computer Security to Teach Computer Science Principles.....	483
Major Gregory White, Ph.D., Captain Gregory Nordstrom (ret), <i>USAF Academy</i>	
U.S. Government Wide Incident Response Capability.....	489
Marianne Swanson, <i>National Institute of Standards and Technology</i>	
MLS DBMS Interoperability Study	495
Rae K. Burns, <i>AGCS, Inc.</i>	
Yi-Fang Koh, <i>Raytheon Electronic Systems</i>	
MISSI Compliance for Commercial-Off-The-Shelf Firewalls	505
Michael Hale, Tammy Mannarino, <i>National Security Agency</i>	
Designing & Operating a Multilevel Security Network Using Standard Commercial Products.....	515
Richard A. Griffith, Mac E. McGregor, <i>Air Force C4 Technology Validation Office</i>	
Real World Anti-Virus Product Reviews and Evaluations - The Current State of Affairs.....	526
Sarah Gordon, Richard Ford, <i>Command Systems, Inc.</i>	
Security Proof of Concept Keystone (SPOCK).....	539
James McGehee, <i>COACT, Inc.</i>	

Use of a Taxonomy of Security Faults	551
Taimur Aslam, Ivan Krsul, Eugene H. Spafford, <i>Purdue University</i>	
Protecting Collaboration.....	561
Gio Wiederhold, Michel Bilello, <i>Stanford University</i>	
Vatsala Sarathy, <i>Oracle Corp.</i>	
XiaoLei Qian, <i>SRI International</i>	
Design and Management of a Secure Networked Administration System: A Practical Solution	570
Vijay Varadharajan, <i>University of Western Sydney, Australia</i>	
Information Warfare, INFOSEC and Dynamic Information Defense	581
J.R. Winkler, C.J. O'Shea, M.C. Stokrp, <i>PRC Inc.</i>	
Security for Mobile Agents: Issues and Requirements.....	591
William M. Farmer, Joshua D. Guttman, Vipin Swarup, <i>The MITRE Corporation</i>	
Extended Capability: A Simple Way to Enforce Complex Security Policies in Distributed Systems	598
I-Lung Kao, <i>IBM Corporation</i>	
Randy Chow, <i>University of Florida</i>	
IGOR: The Intelligence Guard for ONI Replication.....	607
R.W. Shore, <i>The ISX Corporation</i>	

Invited Papers

Management & Administration

Track F

Ethical and Responsible Behavior for Children to Senior Citizens in the Information Age.....	620
Gale S. Warshawsky, <i>International Community Interconnected Computing eXchange</i>	

Legal Perspectives

Track E

Privacy Rights in a Digital Age	630
William Galkin, Esq., <i>Law Office of William S. Galkin</i>	

Panels

Criteria & Assurance

Track A

Trust Technology Assessment Program	643
Chair: Tom Anderson, <i>National Security Agency</i>	
Panelists:	
Pat Toth, <i>National Institute of Standards and Technology</i>	

Alternative Assurance: There's Gotta Be a Better Way!	644
Chair: Douglas J. Landoll, <i>Arca Systems, Inc.</i>	
Panelists:	
John J. Adams, <i>National Security Agency</i>	
TBD, <i>WITAT System Analysis & Operational Assurance Subgroup Chair</i>	
M. Abrams, <i>The MITRE Organization, WITAT Impact Mitigation Subgroup Chair</i>	
TBD, <i>WITAT Determining Assurance Mix Subgroup Chair</i>	
Certification and Accreditation - Processes and Lessons Learned.....	646
Chair: Jack Eller, <i>DISA, CISS (ISBEC)</i>	
Viewpoints:	
The Certification and Accreditation Process Handbook For Certifiers.....	647
Paul Wisniewski, <i>National Security Agency</i>	
Standards in Certification and Accreditation	648
Candice Stark, <i>Computer Science Corporation</i>	
The Certification of the Interim Key Escrow System.....	652
Ray Snouffer, <i>National Institute of Standards and Technology</i>	
Lessons Learned From Application of the Department of Defense Information Technology	
Security Certification and Accreditation	653
Barry C. Stauffer, <i>CORBETT Technologies, Inc.</i>	
Firewall Testing and Rating.....	655
Chair: J. Wack, <i>National Institute of Standards and Technology</i>	
The Trusted Product Evaluation Program: Direction for the Future	656
Chair: J. Pedersen, <i>National Security Agency</i>	
Common Criteria Project Implementation Status	657
Chair: E. Troy, <i>National Institute of Standards and Technology</i>	
Panelists:	
Lynne Ambuel, <i>National Security Agency</i>	
Murray Donaldson, <i>Communications-Electronics Security Group, UK</i>	
Robert Harland, <i>Communications Security Establishment, Canada</i>	
Klaus Keus, <i>BSI/GISA, Germany</i>	
Frank Mulder, <i>Netherlands National Communications Security Agency</i>	
Jonathan Smith, <i>Gamma Secure Systems, UK</i>	
Developmental Assurance and the Common Criteria.....	660
Chair: M. Schanken, <i>National Security Agency</i>	
Panelists:	
S. Katzke, <i>National Institute of Standards and Technology</i>	
E. Troy, <i>National Institute of Standards and Technology</i>	
K. Keus, <i>BSI/GISA, Germany</i>	
Y. Klein, <i>SCSSI, France</i>	

Secure Networking and Assurance Technologies.....	661
Chair: T. Lunt, <i>Defense Advanced Research Projects Agency (DARPA)</i>	
Panelists:	
K. Levitt, <i>University of California, Davis</i>	
S. Kent, <i>BBN</i>	
Viewpoints:	
Secure Mobile Networks.....	663
J. McHugh, <i>Portland State University</i>	
Adaptable Dependable Wrappers.....	666
D. Weber, <i>Key Software</i>	
Generic Software Wrappers for Security and Reliability	667
L. Badger, <i>Trusted Information Systems, Inc.</i>	
Defining an Adaptive Software Security Metric From A Dynamic Software Fault-Tolerance Measure.....	669
J. Voas, <i>Reliable Software Technologies</i>	

Electronic Commerce

Track B

Using Security to Meet Business Needs: An Integrated View From The United Kingdom	677
Chair: Alex McIntosh, <i>PC Security, Ltd.</i>	
Viewpoints:	
Dr. David Brewer, <i>Gamma Secure Systems, Ltd.</i>	679
Nigel Hickson, <i>Department of Trade & Industry</i>	682
Denis Anderton, <i>Barclays Bank PLC</i>	684
Dr. James Hodsdon, <i>CESG</i>	685
Michael Stubbings, <i>Government Communications Headquarters, UK</i>	686
Security APIs: CAPIs and Beyond.....	687
Chair: Amy Reiss, <i>National Security Agency</i>	
Panelists:	
John Centafont, <i>National Security Agency</i>	
TBD, <i>Microsoft</i>	
Lawrence Dobranski, <i>Canadian Communications Security Establishment, Canada</i>	
David Balenson, <i>Trusted Information Systems, Inc.</i>	
Are Cryptosystems Really Unbreakable?.....	691
Chair: Dorothy E. Denning, <i>Georgetown University</i>	
Panelists:	
Steven M. Bellovin, <i>AT&T Research</i>	
Paul Kocher, <i>Independent Cryptography Consultant</i>	
Eric Thompson <i>AccessData Corporation</i>	
Viewpoints:	
The Mathematical Primitives: Are They Really Secure?.....	692
Arjen K. Lenstra, <i>Citibank</i>	

In Depth**Track C**

Best of the New Security Paradigms Workshop.....	693
Chair: T. Haigh, <i>Secure Computing Corporation</i>	
Viewpoints:	
New Paradigms for Internetwork Security	693
J. T. Haigh, <i>Secure Computing Corporation</i>	
The Emperor's Old Armor	694
R. Blakely, <i>International Business Machines</i>	
Position Statement for New Paradigms Internetwork Security Panel.....	698
S. Greenwald, <i>Naval Research Laboratory</i>	
Reactive Security and Social Control.....	701
S. Janson, <i>Swedish Institute of Computer Science, Sweden</i>	
NISS Whitepaper: A New Model of Security for Distributed Systems	704
W. Wulf, <i>University of Virginia</i>	
Series: Public Key Infrastructure: From Theory to Implementation	707
Public Key Infrastructure Technology	
Chair: D. Dodson, <i>National Institute of Standards and Technology</i>	
Panelists:	
R. Housley, <i>Spyrus</i>	
C. Martin, <i>Government Accounting Office</i>	
W. Polk, <i>National Institute of Standards and Technology</i>	
S. Chokani, <i>Cygnacom Solutions, Inc.</i>	
V. Hampel, <i>Hampel Consulting</i>	
Public Key Infrastructure Implementations	
Chair: W. Polk, <i>National Institute of Standards and Technology</i>	
Panelists:	
P. Edfors, <i>Government Information Technology Services (GITS) Working Group</i>	
D. Heckman, <i>National Security Agency</i>	
D. Dodson, <i>National Institute of Standards and Technology</i>	
J. Galvin, <i>CommerceNet</i>	
W. Redden, <i>Communications Security Establishment</i>	
Establishing an Enterprise Virus Response Program	709
Christine Trently, <i>Mitretek Systems</i>	
Data Warehousing I	711
Chair: John Campbell, <i>National Security Agency</i>	
Panelists:	
Jesse C. Worthington, <i>Informix Software, Inc.</i>	
Viewpoints:	
Data Warehousing, Data Mining, and Security: Developments and Challenges.....	711
Dr. Bhavani Thuraisingham, <i>The MITRE Corporation</i>	
Data Warehousing, Data Mining, and the Security Issues.....	716
Dr. John Campbell, <i>National Security Agency</i>	
Data Warehousing II: The Technology	717
Chair: John Davis, <i>NCSC</i>	
Panelists:	
Dr. Bhavani Thuraisingham, <i>The MITRE Corporation</i>	
Dr. John Campbell, <i>National Security Agency</i>	

Internet

Track D

Introduction to Infowarfare Terminology 718
Francis Bondoc, *Klein & Stump*

Information Warfare: Real Threats, Definition Changes, and Science Fiction 725
Chair: Wayne Madsen, *Computer Sciences Corporation*

Panelists:

Martin Hill, *Office of the Assistant Secretary of Defense C3I/Information Warfare*
Frederick G. Tompkins, Matthew Devost, *Science Applications International Corporation*
Scott Shane, *The Baltimore Sun*
John Stanton, *Journal of Technology Transfer*

Security in World Wide Web Browsers: More than Visa cards? 737
Chair: R. Dobry, *National Security Agency*

Panelists:

C. Kolcun, *Microsoft*
B. Atkins, *National Security Agency*
K. Rowe, *NCSA*

Attack/Defense 738
Chair: J. David, *The Fortress*

Panelists:

S. Bellovin, *AT&T*
W. Cheswick, *AT&T*
P. Peterson, *Martin Marietta*
M. Ranum, *V-One*

The Web Series 739

I. The Web – What is it, Why/How is it Vulnerable

II. Securing the Web

Chair: J. David, *The Fortress*

Speaker:

J. Freivald, *Charter Systems, Inc.*
P. Peterson, *Martin Marietta*
D. Dean, *Princeton University*

Legal Perspectives

Track E

Electronic Data: Privacy, Security, Confidentiality Issues 740
Chair: Kristin R. Blair, Esq., *Duvall, Harrington, Hale and Hassan*

Viewpoints:

Virginia Computer Crime Law 741
The Honorable Leslie M. Alden, Judge, *Fairfax County Circuit Court*

Electronic Data: Privacy, Security and Confidentiality 749
Ronald J. Palenski, Esq., *Gordon and Glickson, P.C.*
Steve A. Mandell, Esq., *The Mandell Law Firm*

Monitoring Your Employees: How Much Can You Do And What Should You Do When You Uncover Wrongdoing?.....	800
Steven W. Ray, Esq., <i>Kruchko & Fries</i>	
Computer Crime on the Internet - Sources and Methods.....	817
Chair: Christine Axsmith, Esq. <i>The Orkand Corporation</i>	
Panelists:	
Special Agent Mark Pollitt, <i>Federal Bureau of Investigation</i>	
Phil Reitingner, Esq., <i>Department of Justice</i>	
Barbara Fraser, CERT, <i>Carnegie Mellon University</i>	
Legal Liability for Information System Security Compliance Failures:	
New Recipes for Electronic Sachertorte Algorithms.....	818
Chair: Fred Chris Smith, Esq., <i>Private Practice, Santa Fe, New Mexico</i>	
Panelists:	
John Montjoy Sr., <i>BBN Corporation</i>	
Edward Tenner, <i>Princeton University</i>	
David J. Loundy, Esq., <i>Private Practice, Highland Park, Illinois</i>	
V-Chip: Policies and Technology.....	822
Chair: Hilary Hosmer, <i>Data Security, Inc.</i>	
Panelists:	
D. Moulton, Esq., <i>Chief of Staff, Office of Congressman Markey, HR</i>	
Dr. D. Brody, MD, <i>American Academy of Child and Adolescent Psychiatry</i>	
Ms. S. Goering, Esq., <i>American Civil Liberties Union</i>	
W. Diffie, <i>Sun Microsystems</i>	
Protecting Medical Records and Health Information.....	824
Chair: Joan D. Winston, <i>Trusted Information Systems, Inc.</i>	
Panelists:	
Gail Belles, <i>VA Medical Information Security Service</i>	
Bill Braithwaite, <i>US Department of Health and Human Services</i>	
Paula J. Bruening, <i>Information Policy Consultant</i>	
Patricia Taylor, <i>US General Accounting Office</i>	
Crimes in Cyberspace: Case Studies	827
Chair: William S. Galkin, Esq., <i>Law Office of William S. Galkin</i>	
Panelists:	
Arnold M. Weiner, Esq., <i>Weiner, Astrachan, Gunst, Hillman & Allen</i>	
Kenneth C. Bass, III, <i>Venable, Baejter, Howard & Civeletti</i>	

Management & Administration

Track F

Current Challenges in Computer Security Program Management 828

Chair: Mark Wilson, *National Institute of Standards and Technology*

Panelists:

Lynn McNulty, *McNulty and Associates*

Paul M. Connelly, *White House Communications Agency*

Ann F. Miller, *Fleet and Industrial Supply Center*

Barbara Gutmann, *National Institute of Standards and Technology*

Achieving Vulnerability Data Sharing 830

Chair: Lisa J. Carnahan, *National Institute of Standards and Technology*

Panelists:

Matt Bishop, *University of California, Davis*

James Ellis, *CERT/Coordination Center, Carnegie Mellon University*

Ivan Krsul, *COAST Laboratory, Purdue University*

Incident Handling Policy, Procedures, and Tools 831

Chair: Marianne Swanson, *National Institute of Standards and Technology*

Panelists:

Kelly Cooper, *BBN Planet*

Thomas Longstaff, *Computer Emergency Response Team/Coordination Center*

Peter Richards, *Westinghouse Savannah River Company*

Ken van Wyk, *Science Applications International Corporation*

Interdisciplinary Perspectives on Information Security: Mandatory Reporting 833

Chair: M.E. Kabay, Ph.D., *National Computer Security Association*

Panelists:

Bruce Butterworth, *Federal Aviation Administration*

Barbara Smith Jacobs, *Securities and Exchange Commission*

Bob Whitmore, *Occupational Health and Safety Administration*

Dr. Scott Wetterhall, *Centers for Disease Control and Prevention*

International Perspectives on Cryptography Policy 835

Chair: Dorothy E. Denning, *Georgetown University*

Panelists:

Peter Ford, *Attorney General's Department, Australia*

David Herson, *Commission of the European Communities, Belgium*

Viewpoint:

International Perspectives on Cryptography Policy: A UK Perspective 836

Nigel Hickson, *Department of Trade and Industry, UK*

Security Protocols/Protocol Security 838

Chair: D. Maughan, *National Security Agency*

Surviving the Year 2000 Time Bomb.....	839
Grace L. Hammonds, <i>AGCS, Inc.</i>	

Panelists:

James W. White, *National Director of the Millenium Solutions Center, OAO Corporation*
 Andrew Hodyke, *United States Air Force, ESC/AXS*

Research & Development

Track G

Database Systems Today: Safe Information at My Fingertips?	842
Chair: John R. Campbell, <i>National Security Agency</i>	

Panelists:

Tim Ehram, *Oracle*
 Dick O'Brien, *Security Computing Corporation*
 Thomas Parenty, *Sybase Corporation*
 LTC Ken Pointdexter, *DISA*
 Satpal S. Sahni, *3 S Group Incorporated*

Webware: Nightmare or Dream Come True?.....	844
Chair: Peter G. Neumann, <i>SRI International</i>	

Viewpoints:

Java – Threat or Menace?.....	845
Steve Bellovin, <i>AT&T Research</i>	

Language-based Protection: Why? Why Now?.....	846
Ed Felten, Drew Dean, Dan S. Wallach, <i>Princeton University</i>	

Untrusted Application Need Trusted Operating Systems.....	847
Paul Karger, <i>International Business Machines</i>	

Webware: Widely Distributed Computation Coming of Age	849
James A. Roskind, <i>Netscape Communication Corporation</i>	

Secure Systems and Access Control	851
Chair: T. Lunt, <i>Defense Advanced Research Projects Agency (DARPA)</i>	

Viewpoints:

Domain and Type Enforcement Firewalls.....	852
D. Sterne, <i>Trusted Information Systems, Inc.</i>	

Task-based Authorization: A Research Project in Next-generation Active Security Models .	854
R. Thomas, <i>ORA</i>	

User-centered Security and Adage.....	855
M. Zurko, <i>OSF</i>	

Encapsulated Environments Using the Flux Operating System	857
J. Lepreau, <i>University of Utah</i>	

Facing the Challenge: Secure Network Technology for the 21 st Century	867
Chair: R. Schaeffer, <i>National Security Agency</i>	

Panelists:

R. Meushaw, *National Security Agency*
 C. McBride, *National Security Agency*
 D. Muzzy, *National Security Agency*
 B. Burnham, *National Security Agency*

Toward a Common Framework for Role-Based Access Control	868
Chair: David Ferraiolo, <i>National Institute of Standards and Technology</i>	
Panelists:	
Dr. Ravi Sandhu, <i>George Mason University</i>	
Dr. Virgil Gligor, <i>University of Maryland</i>	
Rick Kuhn, <i>National Institute of Standards and Technology</i>	
Thomas Parently, <i>Sybase</i>	

Solutions

Track H

MISSI Security Management Infrastructure The Certificate Management Infrastructure: Now and In the Next Year	871
Chair: A. Arsenault, <i>National Security Agency</i>	
Panelists:	
D. Heckman, <i>National Security Agency</i>	
S. Capps, <i>National Security Agency</i>	
S. Hunt, <i>National Security Agency</i>	
Future of Trust in Commercial Operating Systems	872
Chair: T. Inskeep, <i>National Security Agency</i>	
Panelists:	
K. Moss, <i>Microsoft</i>	
J. Alexander, <i>Sun Microsystems</i>	
J. Spencer, <i>Data General</i>	
M. Branstad, <i>Trusted Information Systems, Inc.</i>	
G. Liddle, <i>Hewlett Packard</i>	
Vendors Experience with Security Evaluations	873
Chair: Jeff DeMello, <i>Oracle Corporation</i>	
Panelist:	
Janice Caywood, <i>Digital Equipment Corporation</i>	
Viewpoints:	
Duncan Harris, <i>Oracle Corporation</i>	874
Ken Moss, <i>Microsoft Corporation</i>	876
Ian Prickett, <i>Sun Microsystems</i>	877
Workshop Report on the Role of Optical Systems and Devices for Security	879
Chair: Terry Mayfield, <i>Institute for Defense Analyses</i>	
Panelist:	
Mark Krawczewicz, <i>National Security Agency</i>	
Viewpoints:	
Security Issues For All-Optical Networks	882
Muriel Medard, <i>MIT Lincoln Laboratory</i>	
Security for All-Optical Networks	883
Jeff Ingles, Scott McNown, <i>National Security Agency</i>	

Optical Processing Systems for Encryption, Security Verification, and Anticounterfeiting.....	886
Bahram Javidi, <i>University of Connecticut</i>	

Closing Plenary Session

Information Systems Security: Directions and Challenges

Chair: Dr. Willis H. Ware, Corporate Research Staff, Emeritus, *The Rand Corporation*

Panelists:

J. F. Mergan, *BBN*

Stephen Smaha, *Haystack Labs*

Charles Stuckey, *Security Dynamics*

Viewpoints:

Information Security Challenges in the Financial Services Industry	889
--	-----

C. Thomas Cook, *Banc One Services Corporation*

Information Systems Auditing Requirements	890
---	-----

John W. Lainhart IV, Inspector General, *U.S. House of Representatives*

Viewpoint

Willis Ware, <i>The Rand Corporation</i>	895
--	-----

The Next Generation of Cybercriminals	896
---	-----

Chair: Mark Gembicki, WarRoom Research, *LLC*

Panelists:

Jim Christy, *Air Force Office of Special Investigation*

Bill Perez, *Federal Bureau of Investigation*

Doug Waller, *Time Magazine*

19th National Information Systems Security Conference

Author Cross Reference Index

Abrams, Marshall 644
Adams, John J. 644
Alden, Leslie M. 741
Alexander, J. 872
Alves-Foss, Jim. 89
Ambuel, Lynne 657
Anderson, Tom 643
Anderton, Denis 684
Arsenault, A. 871
Aslam, Taimur 551
Atkins, B. 737
Axsmith, Christine 817
Badger, Lee 667
Balenson, David. 687
Bass, Kenneth C., III. 827
Bayuk, Jennifer L. 323
Bell, D. Elliott. 378
Belles, Gail 824
Bellovin, Steven M. 691, 738, 845
Belvin, Frank 67
Bilello, Michel 561
Bishop, Matt. 349, 371, 830
Blair, Kristin R. 740
Blakely, R. 694
Bodeau, Deborah 67
Bondoc, Francis 718
Bowman, Dawn E. 296
Braekin, Stephen H. 181
Bradford, Brian 98
Braithwaite, Bill 824
Branstad, M. 872
Brewer, David. 679
Brody, D. 822
Bruening, Paula J. 824
Burnham, Blaine 867
Burns, Rae K. 495
Burr, William E. 438, 452
Butterworth, Bruce 833
Caelli, William J. 152
Campbell, John R. 716, 717, 842
Capps, S. 871
Carnahan, Lisa J. 342, 830
Casey, Thomas A. 221
Caywood, Janice 873
Centafont, John 687
Cheswick, William 738
Cheung, S. 361

Chokhani, Santosh. 463, 707
Chow, Randy 598
Christy, Jim. 896
Clow, Jack D. 54
Connelly, Paul M. 828
Cook, C. Thomas 889
Cooper, Kelly 831
Coy, Stephen. 428
Crawford, R. 361
Currie, Daniel L., III 194
D'Alotto, Leonard J. 259
David, Jon. 738, 739
Davis, John C. 717
Dean, Drew. 738, 846
DeMello, Jeff 873
Denning, Dorothy E. 691, 835
Devost Matthew 725
Diffie, Whitfield. 822
Dilger, M. 361
Dobranski, Lawrence. 687
Dobry, Rob 737
Dodson, Donna 288, 707
Donaldson, Murray 657
Drake, David L. 276
Dunham, James G. 202
Edfors, P. 707
Ehram, Tim 842
Eller, Jack 46, 646
Ellis, James 830
Epstein, Jeremy 12
Farmer, William M. 591
Felten, Ed 846
Ferraiolo, David. 868
Flahavin, Ellen 26
Ford, Peter 835
Ford, Richard. 526
Foti, Jim 288
Frank, J. 361
Fraser, Barbara 817
Freivald, J. 738
Frineke, Deborah. 89, 410
Gabrielson, Bruce C. 313
Galkin, William S. 630, 827
Galvin, J. 707
Gast, Thomas. 34
Gembicki, Mark 896
Gligor, Virgil 868

19th National Information Systems Security Conference

Author Cross Reference Index

Goering, S.	822	Krawczewicz, Mark	879
Gordon, Sarah	526	Krsul, Ivan	551, 830
Gotfried, Roberta	389	Kuhn, Rick	868
Greenwald, S.	698	Kurth, Helmut	131
Griffith, Richard A.	515	Lainhart, John W., IV	890
Guttman, Barbara	342, 828	Landol, Douglas J.	644
Guttman, Joshua D.	591	Lenstra, Arjen K.	692
Haigh, Thomas	693	Lepreau, J.	857
Hale, Michael	505	Levitt, Karl	361, 661
Hammond, Nicolas	173	Liddle, G.	872
Hammonds, Grace L.	839	Longstaff, Thomas	831
Hampel, Viktor E.	109, 707	Loundy, David J.	818
Harland, Robert	657	Lunt, Teresa	661, 851
Harris, Duncan	874	Madsen, Wayne	725
Heberlein, L. Todd	349, 371	Mandell, Steve A.	749
Heckman, D.	707, 871	Mannarino, Tammy	505
Henning, Ronda	398	Martin, C.	707
Herson, David	835	Mastrorocco, Mike	46
Hickson, Nigel	682, 836	Maughan, Doug	838
Hill, Martin	725	Mayfield, Terry	879
Hoagland, J.	361	McBride, Christine	867
Hodsdon, James	685	McCauley, E. J.	22
Hodyke, Andrew	839	McGehee, James	539
Hosmer, Hilary	822	McGregor, Mac E.	515
Housley, Russ	707	McHugh, John	663
Htoo-Mosher, Naomi	1	McIntosh, Alex	677
Hunt, S.	871	McNown, Scott	883
Ingles, Jeff	883	McNulty, Lynn	828
Inskeep, Todd	872	Medard, Muriel	882
Irvine, Cynthia E.	194	Menk, Charles G., III	76
Jacobs, Barbara Smith	833	Meushaw, Robert	867
Janson, S.	701	Miller, Ann F.	828
Javidi, Bahram	886	Montjoy, John, Sr.	818
Jennings, William T.	202	Morse, Katherine L.	276
Joyal, Paul M.	139	Moss, Ken	872, 876
Kabay, M. E.	833	Moulton, D.	822
Kao, I-Lung	598	Mulder, Frank	657
Karger, Paul	847	Muzzy, D.	867
Katsavos, Panos	233	Nasser, Robert	1
Katzke, Stu	660	Nazario, Noel A.	438, 445, 452
Keller, Sharon	288	Neumann, Peter G.	844
Kent, Steve	661	Nordstrom, Gregory	483
Keus, Klaus	34, 657, 660	O'Brien, Dick	842
Klein, Y.	660	O'Shea, C. J.	581
Kocher, Paul	691	Palenski, Ronald J.	749
Koh, Yi-Fang	495	Parenty, Thomas	842, 868
Kolcun, C.	737	Pedersen, J.	656

19th National Information Systems Security Conference

Author Cross Reference Index

Perc, Bill.....	896	Swanson, Marianne.....	489, 831
Peterson, Padgett	738, 739	Swarup, Vipin	591
Pointdexter, Ken	842	Taylor, Patricia	824
Polk, W. Timothy	438, 452, 707	Tenner, Edward	818
Pollitt, Mark	817	Thomas, R.	854
Prickett, Ian	877	Thompson, Eric.....	691
Qian, XiaoLei	561	Thuraisingham, Bhavani.....	711, 717
Ranum, Marcus	738	Tobat, Daniel L.....	250
Ray, Steven W.....	800	Tompkins, Frederick G.....	725
Razvi, Shaan	67	Toth, Pat	643
Redden, W.	707	Trently, Christine M.	471, 709
Reiss, Amy	687	Troy, Eugene.....	657, 660
Reitinger, Phil.....	817	van Wyk, Ken	831
Richards, Peter	831	Varadharajan, Vijay	233, 570
Roskind, James A.	849	Voas, J.	669
Rowe, Ken.....	737	Wack, John	655
Saghi, Gene	89	Wallach, Dan S.	846
Sahni, Satpal S.	842	Waller, Doug	896
Sandhu, Ravi	868	Ware, Willis.....	895
Sarathy, Vatsala	561	Warsawsky, Gale S.	620
Schaeffer, R.....	867	Weber, K.	666
Schanken, Mary.....	660	Wee, C.	361
Shane, Scott.....	725	Weiner, Arnold M.	827
Sheffield, Richard	215	Weiss, Errol S.....	250
Shore, R. W.....	607	Wetterhall, Scott	833
Sibert, W. Olin	334	White, Gregory	483
Smith, Fred Chris	818	White, James W.	839
Smith, Jonathan	657	Whitmore, Bob.....	833
Snouffer, Ray	26, 652	Wiederhold, Gio	561
Spafford, Eugene H.....	551	Wilson, Mark.....	163, 828
Spencer, J.	872	Winkler, Ira S.	306
Spencer, Raymond	420	Winkler, J.R.	581
Staniford-Chen, S.	361	Winston, Joan D.....	824
Stanton, John.....	725	Wisniewski, Paul	647
Stark, Candice	648	Wood, Bradley J.....	267
Stauffer, Barry C.	46, 653	Woodal, Thomas R.	389
Steffan, William L.	54	Wulf, W.....	704
Sterne, D.	852	Yip, R.	361
Stokrp, M.C.	581	Zerkle, D.....	361
Straw, Julian.....	1	Zunic, Nevenko	1
Stubbings, Michael	123, 686	Zurko, S.....	855

E4 ITSEC EVALUATION OF PR/SM ON ES/9000 PROCESSORS

Naomi Htoo-Mosher
IBM Corporation
522 South Road
Poughkeepsie, New York 12601
email: naomi2@vnet.ibm.com

Robert Nasser
IBM Corporation
522 South Road
Poughkeepsie, New York 12601
email: nasser@vnet.ibm.com

Nevenko Zunic
IBM Corporation
522 South Road
Poughkeepsie, New York 12601
email: zunic@vnet.ibm.com

Julian Straw
Syntegra
Sentinel House
Harvest Crescent
Ancells Park
Fleet Hampshire
GU13 8UZ
United Kingdom

Abstract

This paper will discuss the recently completed evaluation of the Processor Resource/System Manager (PR/SM) on the ES/9000 processors (9021 and 9121) against the Information Technology Security Evaluation Criteria (ITSEC). PR/SM achieved an E4 rating which certifies its use as a secure consolidation platform for combining workloads at different security classifications.

The paper will cover the configuration and use of PR/SM in a secure mode, including the intended environment for use and the intended method for use.

There will be a discussion of the security enforcing functions which were certified as part of the evaluation, and why they are important for anyone interested in consolidating workloads while maintaining a high level of isolation and security.

Key Words

PR/SM, ITSEC, CLEF, LPAR, E4, IBM, Multi-level Security, Syntegra, ES/9000

PR/SM OVERVIEW

IBM ES/9000 systems can be initialized in one of two operating modes: basic mode or logically partitioned mode (LPAR). Processor Resource/Systems Manager (PR/SM) provides the capability that enables the ES/9000 system to be initialized in LPAR mode.

PR/SM is a hardware facility designed to allow the resources of a single physical machine to be divided into one or more distinct, predefined logical machines, each known as a 'logical partition'. Each logical partition can be isolated from all other logical partitions and each is capable of running a separate conventional operating system such as MVS/ESA, VM/ESA, 370-XA, VSE or AIX.

Before the system can be initialized, I/O resources of the overall physical computing system are preallocated by building what is known as an Input/Output Configuration Data set (IOCDS). An LPAR IOCDS defines the logical partitions by name, allocates I/O resources to specific logical partitions and specifies the security characteristics of those I/O resources. The IOCDS can be built so that at no time is any real resource allocated to more than one logical partition. The configuration becomes effective as part of the power-on reset of the hardware.

The remainder of the logical partition's resources are defined by the system operator before activation of the logical partition. These resources include storage size, number of logical processors, scheduling parameters and security controls. Resource definitions take effect upon partition activation and generally stay static while the partition they pertain to is active.

PR/SM's dynamic resource management capability offers customers the opportunity to allocate additional resources to partitions with demanding workloads. Additionally, if a partition completes its workload, unlike a physical processor which would be unused, the partition can be deactivated and additional processing time is made available to the remaining partitions. The dynamic resource allocation capability is also a tremendous benefit to customer installations where frequent configuration changes are a norm.

NEED FOR AN EVALUATION

Many security conscious customers are interested in consolidating workloads to gain benefits such as reduced power, cooling, support staff, floorspace, and software license fees. The consolidation needs can only be satisfied with a platform that has the computing capacity to handle multiple logical partitions, as well as having a very high level of separation and isolation between the partitions. This allows customers to separate users based on need to know or to restrict access to workloads where different security clearances are required. Furthermore, any such product would need to have an independent "stamp of approval" from a security evaluation agency rather than a vendor's claim of providing the proper security functions.

The evaluation process aims to provide such "stamps of approval" by performing independent security evaluations of products. Certification provides assurance to customers that the evaluated product satisfies the claims made about it. An independent body has evaluated the product against an internationally accepted standard and has validated the vendor's claims about the product. We targeted our evaluation at the very demanding and previously unattained E4 level.

We strove to provide our customers and certifiers with a very high assurance that our secure logical partitioning functions would satisfy their most demanding requirements.

EUROPEAN EVALUATIONS - CRITERIA AND EVALUATION SCHEME

Recognizing the need for and the benefits of a common, harmonized criteria, France, Germany, the Netherlands and the United Kingdom set out to develop a single criteria. This became known as the Information Technology Security Evaluation Criteria, or ITSEC. This criteria is the basis for most evaluations conducted in Europe, and was the criteria used for the PR/SM evaluation.

Evaluations conducted in the UK are performed by independent Commercial Licensed Evaluation Facility (CLEF) organizations. CLEF's are the only organizations recognized as being competent to perform security evaluations. Evaluations are carried out as a commercial undertaking by means of a contract entered into between a CLEF and the vendor. Vendors are free to select a CLEF of their choice and to negotiate the costs and timescales. For this evaluation, we chose Syntegra (formerly Secure Information Systems Limited). The evaluation is performed on the security-relevant features of a system or product. These security-relevant features are described in the Security Target document (described later in the paper).

The evaluation consists of two phases: the pre-evaluation consultancy phase (PEC) and the formal evaluation phase. The PEC phase lasts approximately six weeks, while the formal evaluation phase lasts anywhere from three to six months.

The Certification Body, operated jointly by the Department of Trade and Industry (DTI) and the Communications-Electronics Security Group (CESG) in the UK, licenses the CLEF's and provides them with advice on technical matters relating to evaluations. The Certification Body maintains a general oversight of all evaluations undertaken, with each evaluation being assigned a certifier who monitors the progress of the evaluation and is responsible for the subsequent certification activities. Through its oversight of all evaluations, the Certification Body ensures a consistent approach across all evaluations.

PRE-EVALUATION CONSULTANCY (PEC) PHASE

The objective of the PEC is to confirm the appropriate level of assurance for evaluation and to obtain agreement from the Certification Body that the product can attain its target assurance level. The PEC is an optional phase under the ITSEC scheme and is principally a means of reducing the uncertainty associated with evaluation. In the case of PR/SM, being the first E4, the level of uncertainty was very great.

Before entering formal evaluation it was determined to obtain pre-evaluation consultancy. Although not a mandatory feature of the UK Scheme, it was considered essential in this case, given IBM's lack of ITSEC expertise, and the high (and untried) target assurance level of E4. In the ITSEC Scheme a vendor (sponsor) defines the target of evaluation, selecting the specific functions and characteristics of the product which are to be subject to evaluation. The target must include a statement of the threats which the product is intended to defend against. Part of the evaluation process is to determine whether the product counters these threats effectively. Next, an assurance level is selected. Under the ITSEC criteria assurance levels range from E0 (unsatisfactory) to E6 (the highest). In contrast to the Dod Trusted Computer System Evaluation Cri-

teria (TCSEC), there is no direct link between functionality and assurance. Instead the functionality claimed must be related to a stated threat, and a rationale provided.

The objectives of the consultancy were as follows:

- To confirm the feasibility of an E4 evaluation of PR/SM;
- To enable IBM to gain a greater understanding of the requirements for ITSEC E4;
- To prepare a Security Target for PR/SM, including a statement of the threat which the product is intended to counter, the security functionality provided, the intended method of use and the target assurance level;
- To prepare ITSEC Effectiveness documentation;
- To help ensure that the product and documentation (including semi-formal design, test plans and results, development procedures and practices) were suitable for evaluation.

As Syntegra consultants were to provide some of the documentation it was necessary to provide some initial product training. In order to avoid duplication, training was provided for the evaluation team leader at the same time.

The UK Scheme requires that evaluators have no vested interest in the outcome of an evaluation. During pre-evaluation consultancy therefore, they can provide factual information, but cannot give opinions. Given that no one had previously attempted an E4 evaluation, there was a need for interpretations of ITSEC requirements in many areas, and IBM therefore sought consultancy from the CLIFF independent of the evaluation team. This allowed much more freedom for the consultants to express their views and to act on behalf of the sponsor. Among the interpretations required were:

- The nature and extent of the requirement for semi-formal design;
- Requirements for covert channel analysis;
- Documentation of testing.

Results of this work were fed back into the Scheme as precedents.

The Security Target was developed through a series of iterations. The consultants were able to provide their knowledge of the required format and content, and the security enforcing functions (SEFs) were developed by applying known customer requirements to a high level understanding of the product design. E4 demands a rigorous expression of the SEFs using a semi-formal notation, and many detailed refinements were necessary as the consultants' understanding of the product improved. The SEFs must provide a complete and consistent defense against the identified threats, and the development of effectiveness documentation, particularly the suitability and binding analyses, helps to ensure this.

The independent consultants were retained for the duration of the evaluation contract to advise IBM on how to address observations raised by the evaluators. This proved effective, particularly as the evaluation was being carried out in Europe, and communication was therefore limited to weekly conference calls and email.

The PEC is divided into several activities. These activities are an attempt to identify any shortfalls in the deliverables or product and to minimize the risk of failure to achieve certification at the level selected. Some of the PEC activities are:

- *Planning, familiarization and initiation.* It is during this activity that detailed evaluation plans are generated. The evaluation team members become familiar with the security features of the product. Product training for the evaluators may be necessary.

- *Examination of the Security Target.* This document identifies the target assurance level and specifies the security functions of the product which will be evaluated. In addition, the Security Target contains information as to the intended method of use for the product, the intended environment, and the assumed threats in that environment. It will also identify the version of the product which is being evaluated.
- *Examination of the development documentation.* The design documentation needs to be in semi-formal design notation. The documentation needs to provide traceability of the security features from high-level documents down to the module prologues or hardware drawings.
- *Examination of configuration control.* An inspection of the documentation of configuration control procedures will be conducted to ensure that secure distribution procedures and effective change controls exist.
- *Examination of the development environment.* Documentation describing the environment in which the product was developed will be reviewed to ensure that the security of the product was maintained.
- *Examination of testing procedures.* Test plans and procedures along with sample tests and test results will be reviewed to determine the extent of product testing.
- *Production of Vendor Report.* All of the work conducted by the CLIEF during the PEC will be documented in the Vendor Report. This report will identify shortfalls in the deliverables for the target assurance level. The report will also suggest corrective actions if any are required. Finally, the report will also contain a management summary of the results of the Consultancy.
- *Production of the Work Program.* The work to be performed during the formal evaluation will be documented in an Evaluation Work Program. This work will reflect the activities relevant to the target level of assurance. This work program is submitted to the Certification Body which either approves, suggests changes, or disapproves the plan. The formal evaluation cannot begin until after the work program is approved. The work program includes shift estimates, people, and schedules.

At the conclusion of the PEC, the CLIEF produced a Vendor Report, Evaluation Deliverables List, and Evaluation Work Program. Errors which were documented in the Vendor Report did not have to be fixed, but, correcting them improved the chances of a successful evaluation.

Formal Evaluation

The results of the formal evaluation are sent to the Certification Body for subsequent certification. The formal evaluation (like the PEC) is divided into several activities. The evaluation team carries out the work specified in the Evaluation Work Program.

- **Examine the Requirements.** This activity will ensure that the functionality is appropriate to meet the identified threat and intended method of use. It will also ensure that the security target is consistent. The team will also confirm that the Formal Security Policy Model defines the underlying policy enforced by the product and that there is consistency between the Security Target and Formal Model.
- **Examine the Architectural Design.** This activity will validate the separation of the security enforcing and security relevant parts of the product within the design documentation. It will also ensure that the architectural design satisfies the requirements for content and presentation, and will examine the evidence for binding of functionality. The team will check that a semi-formal notation is used in the architectural design to produce a semi-formal description.
- **Examine the Detailed Design.** This activity will examine the detailed design documentation to increase the team's understanding of the design and implementation of the protection mechanisms. The team will check for semi-formal notation, that all components are specified, verify mapping of the security enforcing functions to mechanisms, and validate security enforcing and security relevant component interface descriptions.

- **Examine the Implementation** This activity will examine the correspondence between the source code and tests and the detailed design. The test documentation will be scrutinized for coverage and sample tests will be used to check the overall test results. New tests will be executed to search for errors and vulnerabilities.
- **Examine the Development Environment** This activity will examine the configuration control procedures, change control procedures, implementation languages used, and development security to ensure that appropriate controls are (and have been) in place. A development site visit is required. The team may rebuild parts of the product using configuration control tools.
- **Examine the Operational Documentation & Environment** The objective here is to assess the procedures and guidelines to be followed during the operation of the product to ensure that the security features are not compromised. The Ease of Use Analysis, user documentation, administration documentation, product configuration options and delivery options are also reviewed.
- **Examine the Effectiveness** The objective here is to determine the suitability of the product's security enforcing functions to counter the threats to the security policy of the product.
- **Production of the Evaluation Report** The results of the formal evaluation will be documented in the Evaluation Technical Report. The report will describe the evaluation, its objectives, methods and results, and will contain recommendations regarding certification.

The Evaluation Technical Report is submitted to the Certification Body by the CLIFF. The Certification Body writes a Certification Report and issues a certificate. During the evaluation any errors in the product or in supporting documentation are formally recorded, with a description of the problem, its impact and a recommended solution. These problem reports are sent to both the Certification Body and to the vendor for resolution. The formality of this process ensures that the vendor can monitor and respond to the findings of the evaluators in a timely manner, without waiting for a full report.

Evaluation Experiences

PR/SM was developed in 1987 and enhanced over successive years with new functions and features. The initial requirements for PR/SM were to maintain strict separation between logical partitions and prevent any communications between partitions.

The design team succeeded in meeting the design objectives for PR/SM and it wasn't until mid-1993 that we considered an ITSEC evaluation. Since PR/SM was developed prior to us being familiar with the documentation requirements for an ITSEC evaluation, the majority of our work was spent in creating supporting design documentation.

We hired the Syntegra consultancy group to write the effectiveness documents on our behalf. We felt that this would minimize any interpretation problems between English and American and it would provide us with evaluation documents on a faster schedule. In order for the consultancy team to write the documents, we needed to provide them with comprehensive product education. We accomplished this with on-site training in both the US and the UK. Having the Syntegra consultancy team create the effectiveness documents proved to be very valuable.

While the Syntegra team was creating the effectiveness documentation, we spent our time creating "glue" documentation plus a Trusted Facility Manual. The purpose of the "glue" documentation was to describe where each of the security enforcing components were implemented. Since the evaluation covered both hardware and software, we needed to delineate where specific functions were implemented and the method of documentation. The "glue" documentation that we developed was a matrix which identified the specific SBFs and the associated code modules, microcode, hardware schematics, and test modules. An evaluation at the E4 level requires specific uses of semi-formal design documentation and notation, complete with rules. We developed

these rules and modified all of our prologs, functional specifications, and architecture documents to conform to the rules. Since this evaluation covered PR/SM licensed internal code, hardware schematics, and service processor code, this was not a simple undertaking.

Much of the assurance in the evaluation was gained from examination of the design to assess how the requirements have been implemented, and to ensure that a rigorous approach had been adopted to the development process. The formal evaluation included testing of each of the identified security enforcing functions. Tests were created by Syntegra which sought to defeat the security mechanisms and compromise the enforcing functions. The hands-on test duration was three full weeks. Due to the initial design goals of PR/SM, which were strict isolation and separation, not a single functional problem was discovered with the product.

The PR/SM evaluation raised a number of issues with regard to the ITSEC which required clarification of the requirements. These included the requirement for semi-formal design documentation, and for provision of test results from a mainframe testing environment. An interpretation of the covert channel requirements was also required. As with the TCSEC, the ITSEC process becomes a more certain undertaking as the body of interpretation widens and deepens. At present, the higher evaluation levels are largely uninterpreted, and evaluations above E3 require skilled and experienced personnel.

PR/SM SECURITY ENVIRONMENT

For the purposes of the discussion in the following sections, the System Administrator is defined as any person having access to the hardware system console of the ES/9000 processor.

PR/SM provides a powerful tool for enforcing separation between multiple workloads on a single platform. When separation is to be used in support of confidentiality requirements, it is necessary to create an environment where the hardware is physically secure, and to restrict access to all hardware to authorized personnel only. In addition, the remote support facility must be disabled. The security target applies only to LPAR mode.

A strict separation virtual machine monitor (SVMM) restricts the allocation of resources so that there is absolutely no sharing of objects amongst their clients. Although PR/SM may be configured as an SVMM, it may also be configured to run in a mode where sharing of some resources is permitted.

To be used as a strict separation virtual machine monitor, PR/SM should be configured in the following manner:

- The devices should be configured so that no device is accessible by more than one partition (although they may be accessible by more than one channel path).
- I/O (physical) control units should be within a single partition.
- The System Administrator should not reconfigure a channel path unless all devices and control units are on that path only.
- The System Administrator should ensure that all devices and control units on a reconfigurable path are reset before the path is allocated to another partition.
- No channel paths should be shared between partitions.
- The amount of reserved storage for a partition should be zero.
- The System Administrator should ensure that the total of the initial storage allocations of activated partitions does not exceed the available storage.
- The System Administrator should ensure that the total number of processors and co-processors dedicated to activated partitions is strictly less than the number of available

processors and co-processors, unless there are no shared processors, in which case all available processors and co-processors may be dedicated.

- Dynamic I/O configuration changes should be disabled (i.e. changes require a power-on reset).
- Partitions should be prevented from receiving performance resources that are not allocated to them (no partition should have global performance data control authority).
- At most one partition should have I/O configuration control authority (i.e. no more than one partition should be able to update any IOCDS).
- The System Administrator should ensure that write access is disabled for each IOCDS, unless that IOCDS is to be updated (the current IOCDS should not be updated).
- The System Administrator should verify any changed IOCDS after a power-on reset with that IOCDS, before any partitions have been activated (the System Administrator may determine whether the IOCDS has been changed by inspecting the date of the IOCDS).
- No partition should have cross-partition control authority (i.e. no partition should be able to reset or deactivate another partition).

SECURITY ENFORCING FUNCTIONS:

The following descriptions are a subset of the security enforcing functions which were the target of the E4 PR/SM Evaluation.

Identification and Authentication

- PR/SM will associate a unique identifier with each logical partition in the current configuration. This identifier will be used in determining permitted access to resources.

Access Control

- PR/SM will prevent access to the IOCDS part of a configuration by a user, unless the user is the System Administrator, or the user is a logical partition with I/O configuration control authority.
- PR/SM will prevent access to the reconfigurable part of a configuration by a user unless the user is the System Administrator, or the user is a logical partition and:
 - The logical partition has cross-partition control authority and the access is to deactivate or reset a logical partition, or
 - The access is to deallocate storage or logical processor resources allocated to the partition itself, or
 - The access is to allocate storage or logical processor resources to the partition itself.
- PR/SM can be configured so that at most one logical partition has I/O configuration authority.
- PR/SM can be configured so that no logical partition has cross-partition control authority.
- PR/SM can be configured to prevent the shared use by logical partitions of a channel path, control unit or I/O device.
- PR/SM will permit a channel path to be allocated exclusively to one logical partition either by identifying the channel path as dedicated, or by designating the owning partition as isolated (isolation only applies to the partition's reconfigurable channel paths). PR/SM will prevent the de-allocation of such a channel path from the partition, even when the channel path is offline.
- PR/SM will ensure that a reconfigurable or dedicated channel path is never shared.
- PR/SM will permit a physical processor to be dedicated to a logical partition. PR/SM will ensure that a dedicated physical processor is allocated to only one logical partition, and will

prevent the deallocation of the physical processor whilst the logical processor using it is online and not check-stopped.

- PR/SM can be configured so that no logical partitions have global performance data control authority. In this case, a logical partition will only be able to gather performance data about the resources allocated to it.
- PR/SM will ensure that a storage resource is never shared, and that the amount of storage allocated to a logical partition does not exceed the limit specified in the current configuration.
- PR/SM will prevent the transfer of a message between a logical partition and resources that are not allocated to it, except where the logical partition is explicitly authorised to do so. For example, PR/SM will intercept I/O interrupts that are not for the currently executing logical processor and will present them to the appropriate logical processor.

Accountability

- PR/SM will record in an audit log the security-relevant actions of the System Administrator. These actions are:
 - Creating or modifying the IOCDS part of a configuration.
 - Modifying the reconfigurable part of a configuration.
 - Selecting a configuration to become the next current configuration.
 - Installing a selected configuration by a power-on reset.
 - Activating or deactivating logical partitions.
 - PR/SM will prevent the deletion or modification of these records except when the allocated audit space has been filled. In this case, the system may overwrite old audit records with new audit records in time order (i.e. oldest first).
 - PR/SM will prevent the reading of the audit log by logical partitions.

Object Reuse

- PR/SM will ensure the clearing of information from a storage resource before that resource is allocated to a logical partition.
- PR/SM will ensure that the information in a physical processor or co-processor that is available to the currently executing logical processor is unaffected by any previously executing logical processor from another logical partition. For example, on a context switch, the control registers, general registers and program status word in the physical processor will be restored to their previously saved values.
- PR/SM will send a reset signal to a non-shared channel path and its attached I/O devices before that channel is allocated to a logical partition.

SEF Evolution

We set out with a set of high level objectives in terms of separation of partitions, configuration of the system, etc. These were resolved into precise statements of security functionality which could be tested. The SEFs are also supported by a set of precise definitions of terms used. The issue of completeness and consistency of the SEFs is addressed in effectiveness documentation (principally suitability and binding), which is in turn examined by the evaluators. This part of the process is essential to an evaluation scheme which does not use predefined functionality, as does the TCSEC, and which allows for a wide variety of evaluated products. The evaluation and certification process is designed to avoid any abuse of the freedom which this approach provides to vendors.

SUMMARY

In the case of PR/SM, the E4 certificate provides a very high degree of assurance that PR/SM can be used in environments where separation of workloads is a requirement, but where the use of a single hardware platform is desirable for reasons of economy, flexibility, security or management. PR/SM provides for secure isolation by preventing the flow of information between logical partitions. This isolation may be used where the separation is based on need to know, or where data at differing security classifications must be kept apart. When used in accordance with the evaluation documentation it is capable of providing a multi-level secure consolidation platform. The claims made about PR/SM have been validated.

REFERENCES

1. Information Technology Security Evaluation Criteria (ITSEC) Provisional Harmonised Criteria, June 1991.
2. ES/9000 Reference Guide, G320-9996-00, IBM Corporation, 09-90.
3. PR/SM Planning Guide - Document Number GA22-7123

TRADEMARKS

The following are trademarks of the International Business Machines Corporation in the United States or other countries or both:

AIX

ES/9000

IBM

MVS/ESA

PR/SM

Processor Resource/Systems Manager

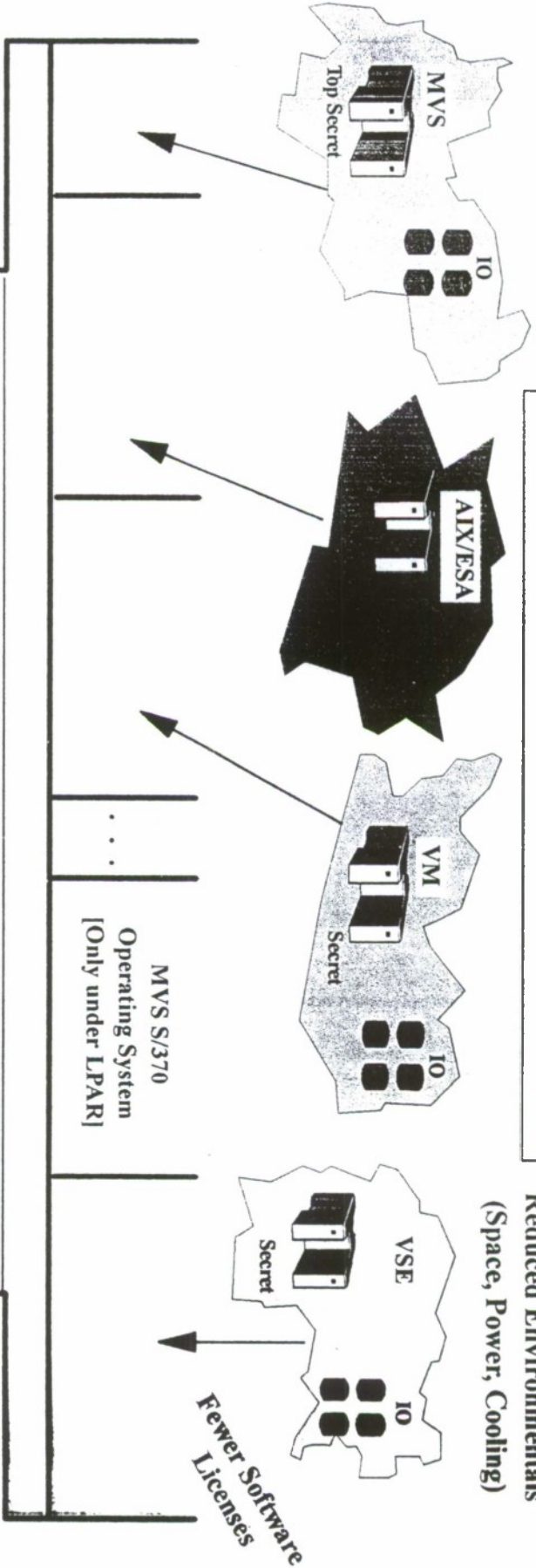
VM/ESA

Reduced Operations
& Support Personnel

ES/9000 PR/SM Secure Consolidation ITSEC Certified Product Level E4

Reduced Environmentals
(Space, Power, Cooling)

Fewer Software
Licenses



Separation &
Isolation

Simplified I/O
Device Management

Flexibility



TCB

Multi-Level Security Operation

Dynamic Workload Balancing

Single System
Look and Feel

BN102095.PRE

A HIGH-PERFORMANCE HARDWARE-BASED HIGH-ASSURANCE TRUSTED WINDOWING SYSTEM

Jeremy Epstein
Cordant, Inc.
11400 Commerce Park Drive
Reston VA 22091
jepstein@cordant.com

Abstract¹

TRW's Trusted X Window System prototype established that it is possible to build a high assurance windowing system, given a trusted operating system as a base. This paper describes an extension of that architecture that uses custom designed hardware to provide a high-performance, low-cost windowing system while retaining the high-assurance character of the original design.

1. Introduction

The TRW Trusted X Window System (henceforth TX) prototype showed that high assurance multi-level secure windowing is not an oxymoron. [TXArch93] describes the TX architecture. However, TX has a fundamental performance limitation: all screen drawing is performed by updating a "virtual frame buffer", which is then merged by the software TCB into the physical screen. As a result, screen updates are slow. In addition, software is unable to take advantage of any graphics hardware, because hardware access is limited to the TCB.

In this paper we describe the design for a hardware board, which coupled with the TX design can yield a high-performance, high-assurance, low-cost workstation.

Section 2 gives a brief introduction to the TX architecture. Section 3 describes the design for the hardware, and contrasts it with both the software-based solution in TX, and with other hardware-based solutions. Section 4 describes some particular considerations in building the proposed board for IBM PC hardware. Section 5 compares this architecture to related work, while section 6 summarizes our results. Section 7 is a summary of acronyms used in the paper.

2. TX Architecture

The TX architecture, as described in [TXArch93], relies on an underlying high-assurance (e.g., TCSEC B3 or A1 [TCSEC85]) operating system that supports both single-level and multi-level subjects over a range of Mandatory Access Control levels with high-bandwidth inter-subject communication. TX uses the time-tested concept of replication (or polyinstantiation) of subjects to allow untrusted software to provide most of the system's functionality. Figure 1 shows a simplified version of the TX architecture.² Items shown above and to the left of the double line are non-TCB, while items shown below and to the right of the double line are TCB elements.

Keyboard and mouse input is received by TX/IM, and passed to the X single level server³ (TX/SLS) corresponding to the currently selected MAC label. That is, if the currently selected label is Secret/B/, then all input would be passed to the X server on the left in Figure 1, and the X server on the right would be unable to detect the presence of any input. The only processing performed by TX/IM is to search for the Secure Attention Key (SAK) sequence, which is used to invoke the trusted path facility.

An unmodified X client makes requests to the TX/SLS running at its same MAC label, which in turn draws into a single level *virtual frame buffer* (VFB). Each VFB represents what a single TX/SLS views as the contents of the screen. When a TX/SLS updates its VFB, it notifies TX/DM, which merges the VFBs for all TX/SLSs and updates the *physical frame buffer*

²The following items are omitted: TX/PE and TX/SEs (used for cut and paste support); TX/CIT, TX/SIT, and TX/M (used for initialization).

³In X, the term *server* refers to the software that manages the graphics hardware, while a *client* is an application that uses graphics hardware. Thus, an X server is always local to a user, while a client may be local or remote. While consistent, this nomenclature frequently confuses new users of X, who are used to clients being local and servers being remote!

¹Copyright © 1996 Cordant, Inc. All Rights Reserved.

(PFB), which is used by the graphics hardware to update the screen. Access to the PFB is limited to TX/DM, along with access to all other graphics hardware (because the graphics hardware is not trusted to provide separation of requests from multiple MAC labels).

Users can manipulate only those windows at the current selected MAC label. That is, to move, resize, or provide keyboard or mouse input to a window, the current MAC label must equal the window's MAC label. This is enforced by TX/IM, which sends all input to the TX/SLS corresponding to the currently

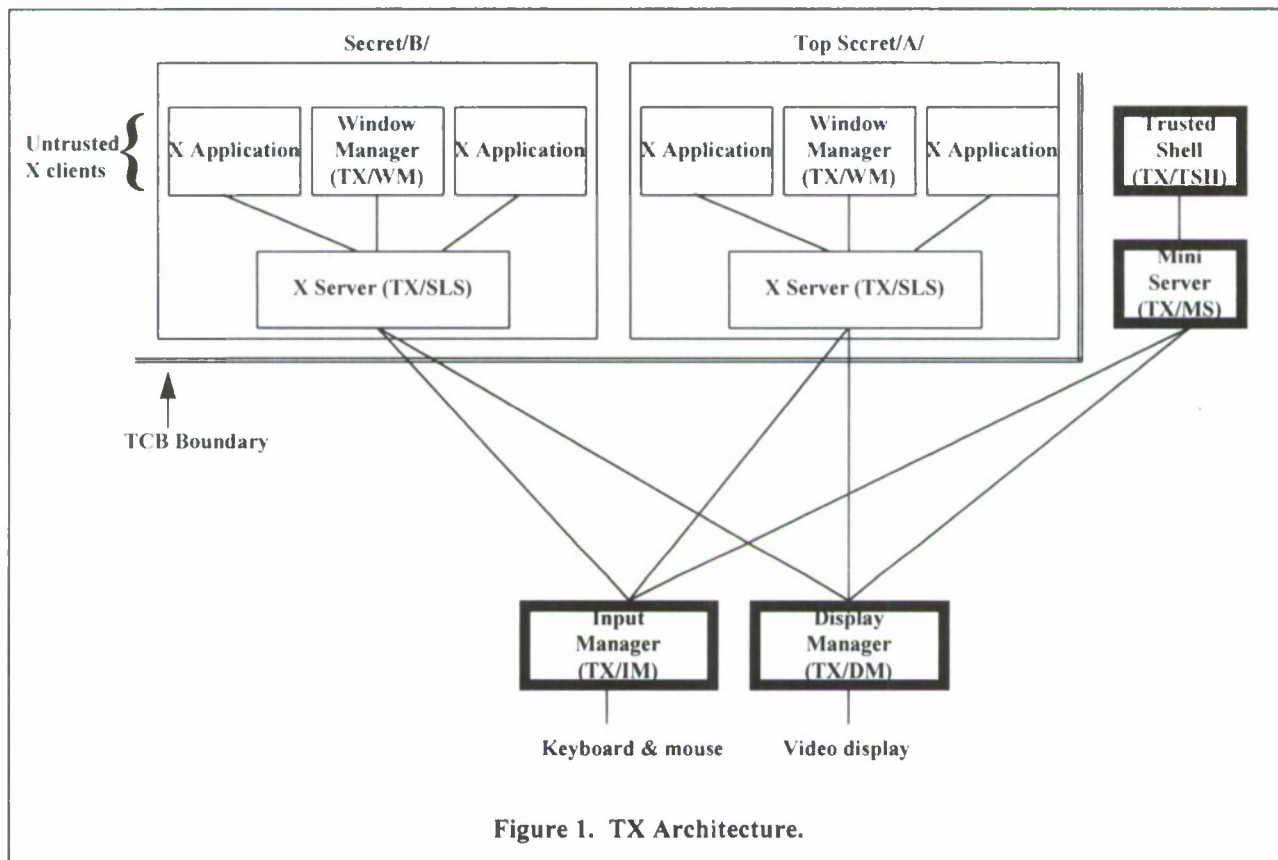


Figure 1. TX Architecture.

When an application action causes mapping or unmapping of a window⁴ on the screen (typically associated with starting a new application), the corresponding TX/SLS sends a message to TX/DM, which in turn recalculates the screen layout based on the new set of windows on the screen. When mapping windows, TX/DM also draws visible labels representing the MAC label on all four sides of the window.

Each TX/SLS has a matching TX/WM that performs (untrusted) window management of windows at that MAC label.

⁴Strictly speaking, this process only occurs with top level windows, as windows are defined in a hierarchical fashion in X (i.e., each push-button is a window within a pane of push-buttons, which is a window within a dialog box, which is a window within a top-level window).

selected label.

TX/MS performs an analogous function to the TX/SLS for the trusted shell application (TX/TSH). That is, it allows TX/TSH to draw on the screen using a small set of graphics primitives. TX/MS draws in a VFB, just as the TX/SLSs do. TX/DM merges the VFB belonging to TX/MS with the TX/SLS's VFBs, although TX/MS always takes precedence. Functions provided through TX/TSH include changing the current MAC label and starting instances of TX/SLS at new MAC labels. TX/MS and TX/TSH are inactive, except when invoked by the user through the SAK (as described above under TX/IM).

Figure 2 shows a sample window display, with shading to indicate which portion of the TX system provides the information on the screen. The background portion of the screen contains a fill pattern that is selected by the lowest TX/SLS (i.e., the least highly classified) associated with the login session.

Note that there is a complete set of processing software (i.e., TX/SLS, TX/WM, and applications) with a corresponding VFB for each unique MAC label in use. Thus, if a user is concurrently working with data at four different classification levels, there would be four X servers, four window managers, and four sets of applications running. Each unique combination of non-hierarchical categories is considered a different MAC label for purposes of TX software replication.

the processing problems, provides a brief introduction to graphics hardware, and proposes a solution based on hardware polyinstantiation.

3.1. The Problem

Consider how a character typed by a user is echoed to the screen in an ordinary X system as compared to TX.⁶ In ordinary X, the X server receives the character and routes it to the appropriate X client. That client

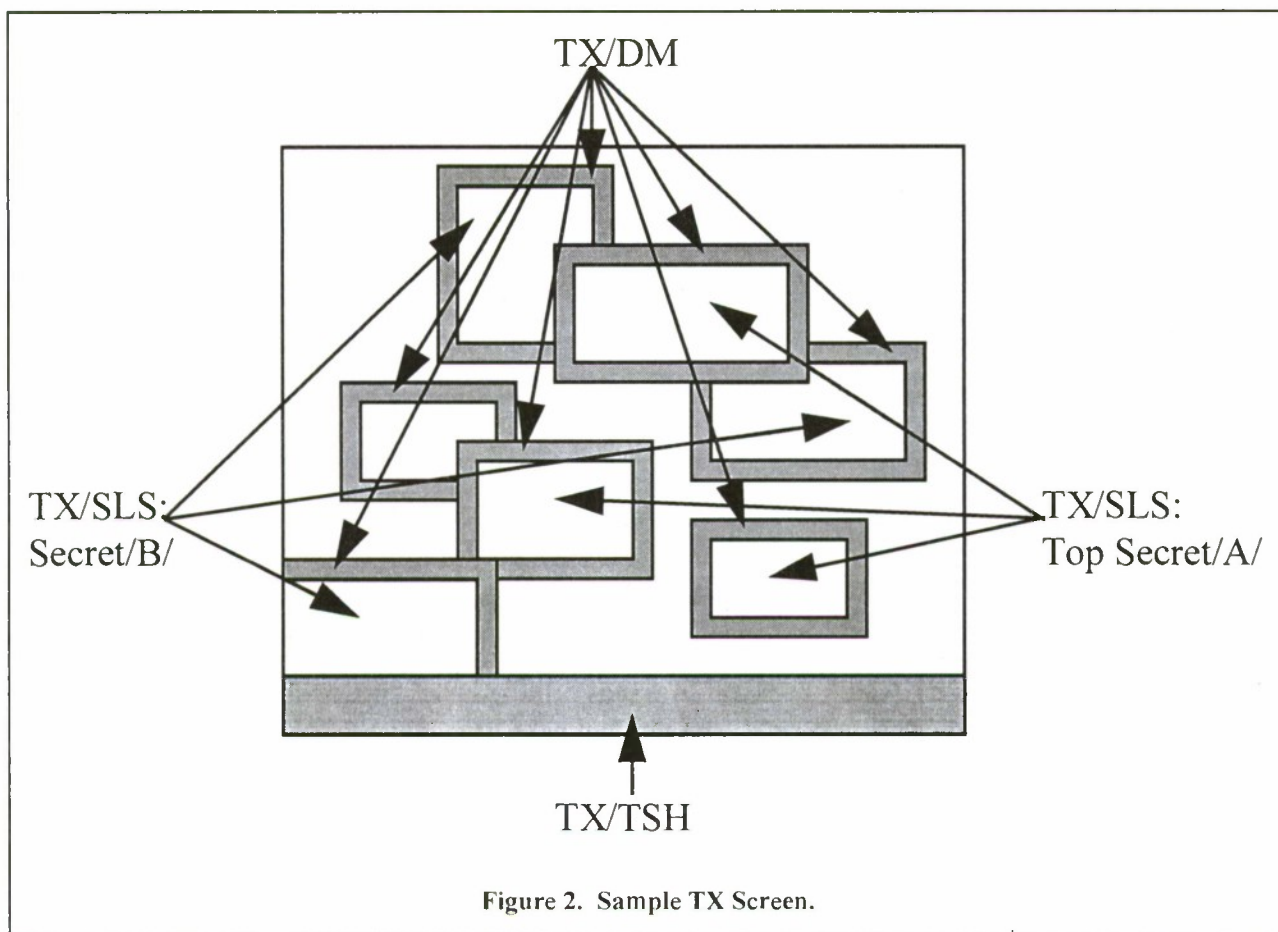


Figure 2. Sample TX Screen.

3. Hardware Design

There are several performance problems with the TX architecture relating to output processing due to the extra level of processing involved with each change to the display. While input processing theoretically has the same performance problem (due to TX/IM having to examine each keystroke for the SAK), human input rates are low enough that the additional overhead is not noticeable⁵. The remainder of this section describes

responds by instructing the X server to draw the character in the window. The X server verifies that the portion of the window being drawn is not obscured by another window (or off the screen, or otherwise unavailable), and renders the character into the PFB, possibly using a hardware assist (i.e., hardware capable of rendering characters from a font stored memory). By contrast in TX, TX/IM receives the character, routes it to the X server (i.e., TX/SLS) at the currently

⁵While the input processing speed was not noticable, the resources required to echo characters to the screen was quite noticable in the prototype implementation.

⁶Character processing in X is full-duplex: the X server does not provide echo, but rather relies on the X client to perform the echo. This is necessary because only the X client knows where to draw the character, what font to use, etc.

select MAC label, which routes the character to the appropriate X client. Again, the client responds to the X server (i.e., TX/SLS), which performs the same processing, but draws the character into its VFB *without* any hardware assist. TX/DM then verifies that the portion of the VFB being updated is not obscured by a window at a different MAC label (a given TX/SLS is unable to determine this, since it only knows about windows at its own MAC label), and if no such limitation exists, copies the character from the VFB of the TX/SLS to the PFB, thus causing its display on the screen.

In a more extreme case, consider where an action would cause an X client to perform a three-dimensional rotation of an image in a screen, or where video is being shown in a window. In ordinary X, the client can request that the X server perform a 3D rotation (providing that the server provides that facility), which can be done in hardware. Similarly, a client can request that the server show a video in a window (again, providing that the server provides the facility), which may be possible in hardware. In TX, these features cannot use hardware, because the TX/SLS has no direct hardware access. Hence, for such real-time and high-performance graphics, TX is too slow to be usable.

Thus, we note that a system using TX as its multi-level secure windowing system will be unable to take advantage of high performance graphics hardware, and will hence be limited in its graphics performance. In addition, the context switching and message passing time in the underlying operating system becomes critical to the performance of the user interface.

3.2. A Brief Introduction to Graphics Hardware

To understand the proposed solution, it is necessary to have a high-level view of how graphics hardware works. As previously noted, ordinary X servers draw into a PFB. A PFB is a one-dimensional array, where each element of the array represents a single pixel on the screen. Each element of the PFB is one or more bits, depending on the type of video being drawn and the cost. For example, a black and white monitor would be driven by a PFB with one bit per pixel (where the definition of whether 0 represents black or white is dependent on the hardware designer). For color or grayscale graphics controllers, common values are four bits per pixel (16 simultaneous color or grayscale values possible), eight bits per pixel (256 simultaneous color or grayscale values possible), and 24 bits per pixel (16M simultaneous color or grayscale values possible).

It is thus possible to calculate the memory requirements for a PFB by multiplying the resolution to be provided (e.g., 1024 x 768) by the number of bits per pixel. Graphics controllers for current model IBM PCs typically have 1MB, which allows for 1024 x 768 with eight bits per pixel.

For four or eight bit controllers, the pixel value is not usually a color definition *per se*, but rather an index into a *colormap* which selects the red, green, and blue values associated with pixels having that value. Thus, it is possible to recolor all pixels of a single value without modifying the pixels, but rather by modifying the colormap entry. Depending on the graphics hardware, the colormap may be managed directly by the X server or by the operating system. If the operating system manages the colormap, then the X server uses system calls to request changes to the map. For 24 bit controllers, the pixel value typically contains eight bit red, green, and blue values, and hence no colormap is needed.

For maximum performance, X servers map the PFB into their address space and manipulate the PFB directly using ordinary memory load and store instructions. That is, the PFB is not managed by the operating system's kernel, as the overhead in making operating system requests to modify each pixel would render the hardware unusable. Low-end graphics hardware converts the values in the PFB together with the colormap into electrical signals to the monitor. In this case, the X server translates high-level requests (e.g., draw a three-pixel wide line from pixel (X1,Y1) to (X2, Y2)) into modifications to pixels within the PFB. More sophisticated graphics hardware may include facilities to offload such drawing, so the host processor can spend more cycles running the application itself. High-end graphics hardware can include a buffer of related commands, and can then perform tasks such as rotation without any involvement by the X server.

In X, clients do not directly manipulate either the PFB or the colormap, but rather rely on the X server to perform those tasks.

3.3. Proposed Solution

Our goal is to reduce the performance bottleneck caused by TX/DM having to mediate all access to the PFB. To do so, we propose a hardware solution with multiple *Virtual Graphics Subsystems* (VGS),⁷ each with its own VFB. Each TX/SLS would then have

⁷Each graphics subsystem would typically consist of a single graphics chip. More sophisticated graphics subsystems might use several chips, but that has no impact on our architecture.

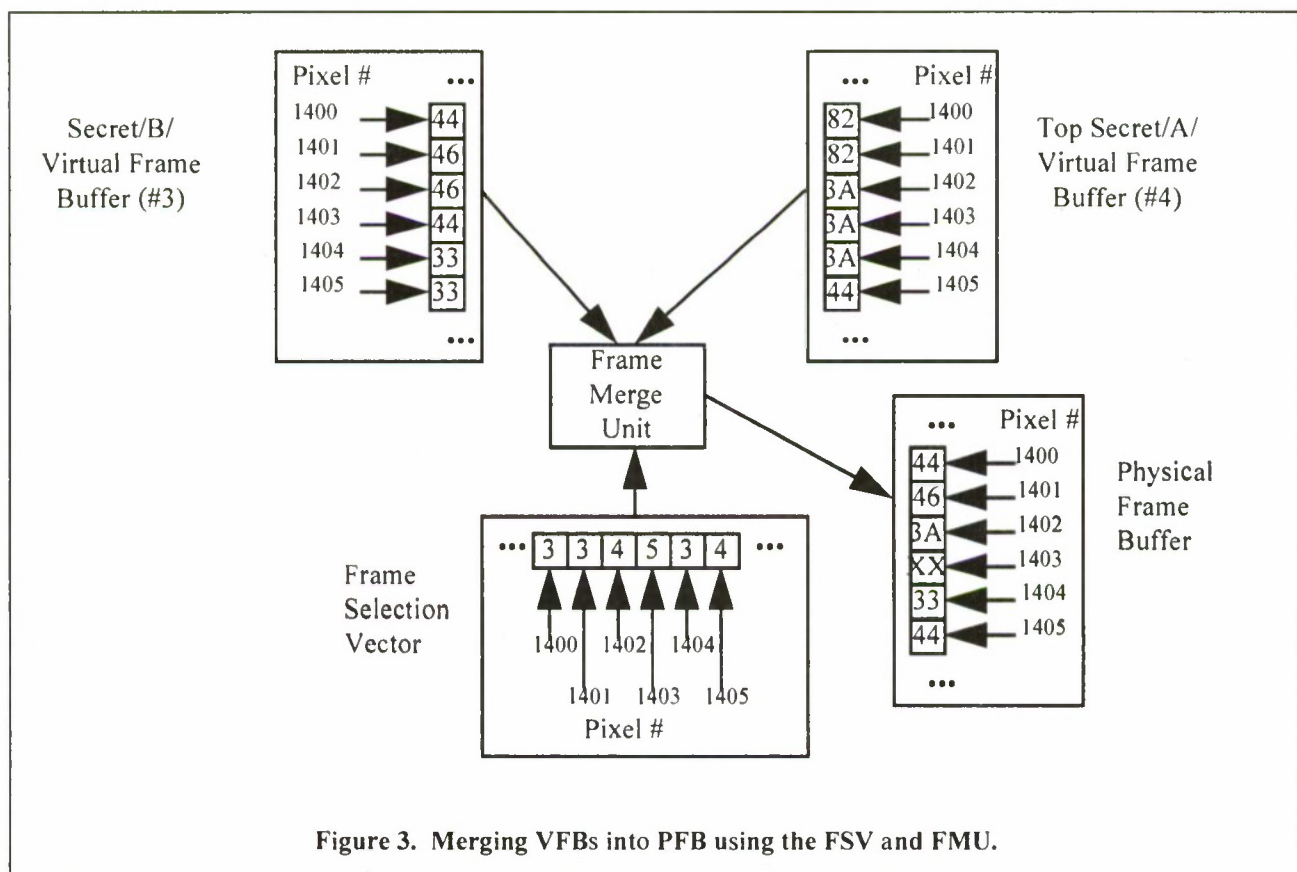
direct access to a VGS, and hence would not need to request that TX/DM perform screen updates. A *Frame Selection Vector* (FSV) determines, for each pixel on the screen, which of the VFBs "owns" that pixel. The *Frame Merge Unit* (FMU) constantly scans the FSV, selecting the pixel from the corresponding VFB and copying it to the PFB. Existing hardware can then translate the PFB into signals to the monitor, just as this operation occurs in current graphics controllers. Thus, if an SLS updates its VFB, the contents of the frame buffer would be updated at the next scan of that pixel (which typically happens 60 or 70 times per second).

Figure 3 shows that for pixel 1400, the FMU selects the value 44, because the FSV entry for pixel 1400 is 3, indicating the Secret/B/ VFB, while for pixel 1402 the FMU selects the value 3A, because the FSV entry for that pixel is 4. The value XX is shown for pixel 1403 to indicate that the value of VFB #5 is not shown in the figure.

clearing the associated VFB. The zeroeth VGS is reserved for use by TX/DM and TX/MS for displaying TCB data, such as the trusted path menus and visual window labels.

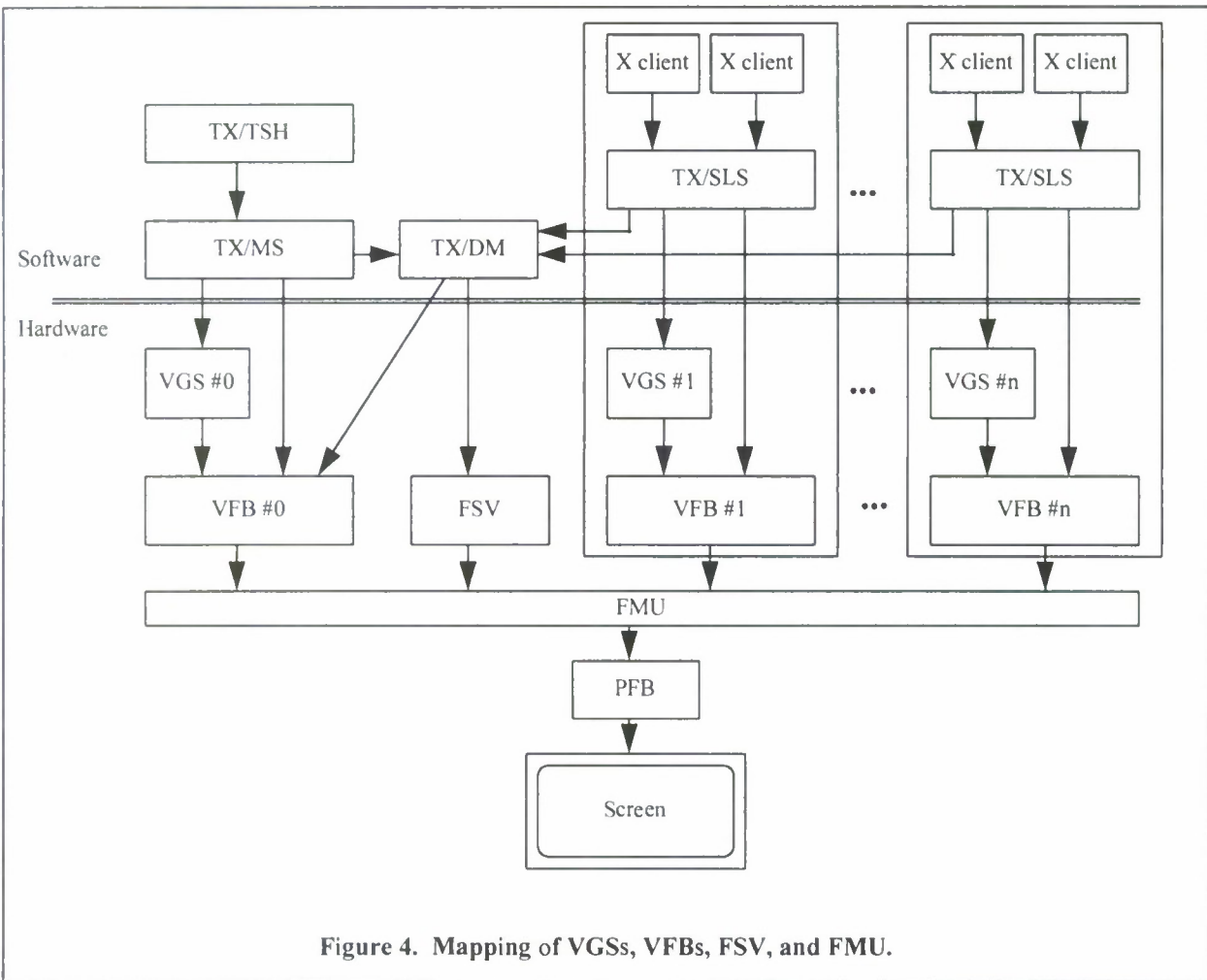
Figure 4 shows the mapping of VGSs, VFBs, the FSV, and FMU. TX/DM still plays an important role in display: it is responsible for (a) drawing labels in windows by rendering the labels into VFB #0 and (b) updating the FSV whenever a window is mapped or unmapped to assign the necessary pixels to the corresponding VGS and VFB. However, these are both events that occur relatively infrequently compared to screen updates. Hence, the performance-critical portion of TX/DM is moved into hardware.

Note that the size of an entry in the FSV need not be the same as the size of a pixel. The FSV entries would typically be four bits, thus allowing up to 15 VGSs plus the reserved VGS for use with TCB data. A VFB entry would typically be eight or 24 bits, as noted above. Note that all VFBs (and the PFB) must be



In this architecture, VGSs and their corresponding VFBs are dynamically allocated when users request creation of new X servers using the TX/TSH. To avoid improper object reuse, there must be a mechanism in the hardware to cause resetting a particular VGS and

identical in size for this scheme to work, as the FMU does not map different size VFB entries into the PFB.



As previously noted, the pixel value in the VFBs and PFB is (typically) an index into a colormap. For example, in Figure 3, the value 44 appears as a pixel value in both VFB #3 (labeled Secret/B/) and VFB #4 (labeled Top Secret /A/). In the TX prototype each TX/SLS configures its colormap independently to avoid the covert channels which might be present if they shared a common colormap. As a result, whenever the user selects a new current MAC label, the colormap from that TX/SLS is installed as the current colormap. The result of this design is that windows change colors in a distracting fashion when the user changes MAC labels. An alternate solution which could have been implemented in TX is to divide up the colormap, allocating certain entries to each TX/SLS in a static fashion. However, this reduces the maximum number of entries allocated to any individual TX/SLS.

In the proposed hardware architecture, if the FMU directly drives the screen, instead of generating a PFB, then it could take the colormaps corresponding to each VFB and directly generate the necessary signals to the

monitor. Alternately, there could be a system colormap for use by the FMU. TX/DM would set the system colormap from the VGS's colormap whenever a MAC label is selected. This latter approach is the equivalent of the prototype TX implementation.

An important premise is that each TX/SLS is capable of communicating with the VGS and VFB at its MAC label, and that there is no mixing of information. The specifics of how this can be implemented depend on both the operating system and specific hardware architecture, and as such are discussed in the following section.

4. A PC Realization

The IBM-compatible personal computer has become the *de facto* standard for workstation hardware. Unfortunately, it has several significant flaws which make an implementation of the above architecture difficult. In this section we outline how the hardware described in section 3 could be implemented in a PC hardware environment. The PC

is probably the most complicated environment to design for; other hardware architectures would be easier than that described here.

shows a block diagram of a PC graphics card designed to use off-the-shelf VGA chips and memory. A *Bus Decode Unit* (BDU) decodes a 16 bit I/O address,

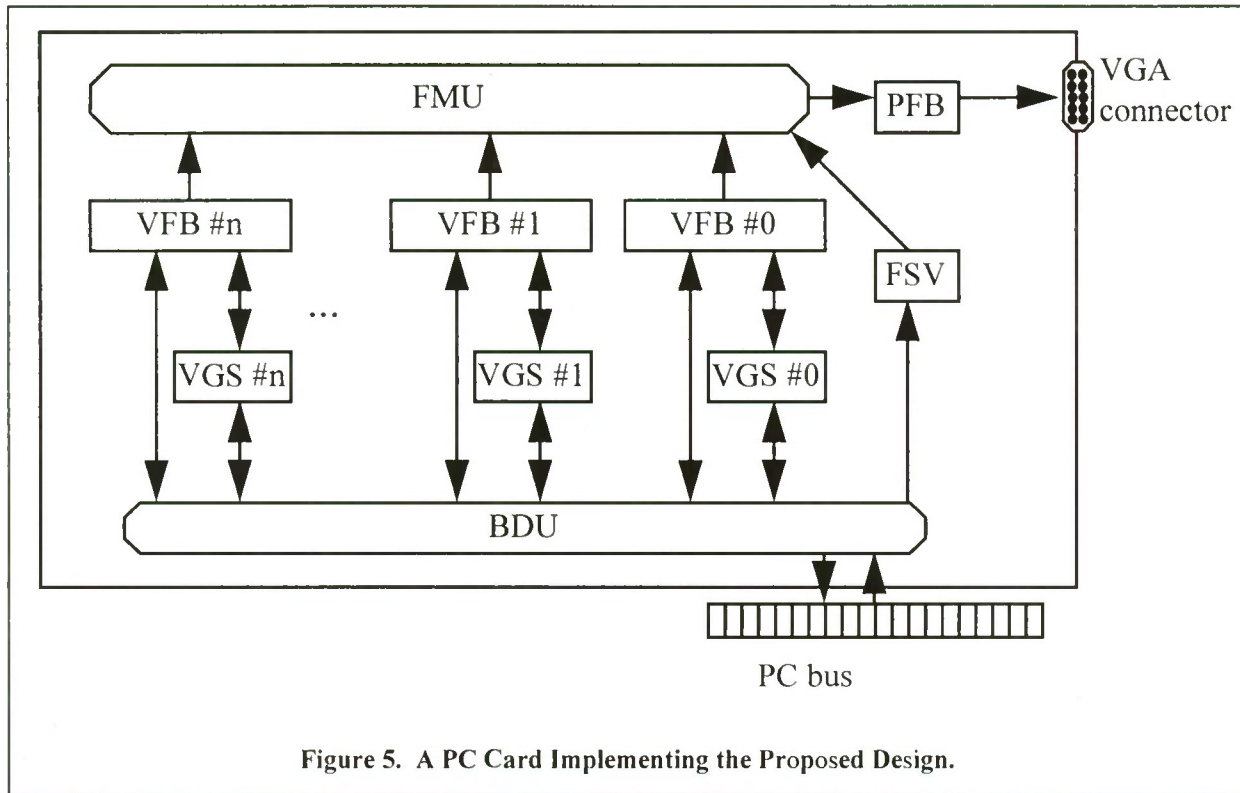


Figure 5. A PC Card Implementing the Proposed Design.

Intel x86 processors (including the 80486 and Pentium) access I/O devices both by issuing IN and OUT instructions and by accessing memory-mapped devices. For example, typical VGA⁸ controllers decode I/O addresses 3C0h through 3CFh for use in a wide variety of operations including setting up colormaps, configuring screen resolution, etc. The PFB for a VGA controller is accessed through a 64K sliding window mapped into main memory.⁹

In order to achieve reasonable performance and minimize cost of the resulting board, it is necessary to have hardware that operates within these limits (i.e., it behaves as a conventional VGA controller). Figure 5

examines the high order six bits, and routes it to the appropriate VGS. The BDU also decodes memory addresses to allow access to the VFBs and the FSV. The BDU relies on the host processor to prevent any invalid access, such as by a TX/SLS to the FSV.

In this design, the VGSs are off-the-shelf SVGA chips, and the VFBs are off-the-shelf memory chips. The PFB is a standard video RAM, and the FSV is an off-the-shelf memory chip. Off-the-shelf components can also be used to translate the PFB to video signals. The only custom hardware is the FMU and the BDU. From our experience in designing other PC hardware, we believe that the FMU and BDU could be combined into a single custom chip (most likely an ASIC, or Application Specific Integrated Circuit). The BDU is moderately complex, as it needs to forward requests from the bus to the appropriate VGS and send responses back, mapping I/O addresses in the process.

Operating system support necessary for this card is as follows: Each TX/SLS would need to have I/O access allowed to the I/O addresses of the corresponding VGS,¹⁰ and its memory map would

⁸VGA, or Video Graphics Array, and its superset Super VGA (or SVGA) is the standard for modern PC video controllers. There are a wide variety of largely incompatible SVGA controllers, each of which use a separate command set.

⁹The 64K window size is a holdover from the early PC architecture, where addresses above 640K were reserved. To move the sliding window, X servers issue OUT commands to the I/O ports, causing the VGA controller to adjust the video addresses mapped into main memory.

¹⁰It is undesirable to require that I/O access to the VGS go through the operating system, as the aforementioned

need to include the corresponding VFB. Fortunately, the Intel 80x86 architecture includes a facility to allow non-ring 0 processes (i.e., those processes not running in the most privileged processor state) to access selected I/O addresses. Thus, the operating system would simply configure the appropriate set of VGS addresses for each TX/SLS, and the CPU hardware performs the necessary protection.

Changes to the X server would be minimal to use this hardware. TX/DM would need a mechanism to notify the TX/SLS of the I/O and memory addresses it should use, rather than the default values hardwired into the code.

4.1. Areas for Future Work

Because we have not performed a detailed hardware design, it is likely that there are flaws yet to be discovered. One item we do not yet have a solution for is how to reliably clear the VGS so it can be dynamically assigned to a new TX/SLS without fear of object reuse. A fallback position would be to either statically assign the VGSs (i.e., VGS #1 is always Top Secret/A/, even if a particular user is not using that MAC label), or to require that the workstation be power-cycled before reassigning VGSs to new labels. Another alternative would be to rely on a hardware feature of the VGSs to perform the clearing on command from the BDU. However, we are unsure whether off-the-shelf VGA chips that would be used for VGSs would have such a facility.

This design allows untrusted software to directly access the bus (to access the VGS and VFB). As a result, it clearly has opportunities for hardware level covert timing channels. We do not have any solution to this problem.

4.2. Estimated Manufacturing Cost

It is not obvious how many VGSs and VFBs a user would need, as different users will need to operate with varying numbers of simultaneous MAC labels. In addition, if VGSs are statically assigned (as noted above might be desirable to avoid covert channels), then more VGSs might be required.

As a result of this uncertainty, we believe that the board described above should be built with three VGSs, three VFBs, and sockets to insert additional VGSs and VFBs. That is, the base model would allow

sliding window is manipulated by performing direct I/O operations to the VGS. Requiring each such operation to go through the operating system would seriously damage performance.

operation of two simultaneous labels, plus a VGS and VFB for use by the TCB for labeling and trusted path.

The actual cost of a board depends on the size of each VFB, the sophistication of the VGSs, and other factors to be determined during detailed hardware design. We have presumed a low-end VGS and 1MB VFBs. We have also assumed that the screen has no more than 1 million pixels, which requires a FSV of 512KB.

Given such assumptions, we believe that such a board could be manufactured today in large quantities (10,000 units) for about \$300 each, as shown in Table 1. Fluctuations in memory cost will obviously affect the price significantly. Adding additional VGS/VFB pairs would cost about \$65 each. Thus, a full board with 16 VGSs and VFBs could be manufactured for $\$300 + (\$65 \times 13) = \$1145$. Note that these figures do not include any allowance for hardware engineering, software development, or profit. While this is certainly not a low-cost board compared to a standard VGA card, it is truly inexpensive compared to having 15 computers on a user's desk!

Table 1. Estimated board manufacturing cost.

Item	Cost
Printed circuit board	\$35
BDU/FMU ASIC	\$20
Physical Frame Buffer (1MB video RAM)	\$40
Virtual Graphics Subsystem (qty 3)	\$75
Virtual Frame Buffer (1MB RAM, qty 3)	\$120
Sockets for 13 more VGSs and VFBs	\$5
Miscellaneous components	\$5
Total (with 3 VGS/VFB pairs)	\$300
Total (with 16 VGS/VFB pairs)	\$1145

4.3. Performance

The design proposed here is such that graphics performance should be almost equal to that of X using a VGA card with the same graphics chip as is used in the VGS. The FMU should not introduce any noticeable overhead. The BDU should not introduce significant overhead either, except when multiple instances of TX/SLS are contending for bus access to access the BDU. Using a fast bus (e.g., a PCI bus) should minimize such contention.

Operating system overhead is a significant concern, as context switching and message passing times can cause software bottlenecks. However, the performance of this design should be significantly better than that of the TX prototype which sends the VFB from the

TX/SLS to the TX/DM in a message, which is a significant strain on operating system message passing.

5. Related Work

The closest work to that described here is a patent granted to Loral [Loral91]. The notion of using polyinstantiated hardware is common to the approach presented there. However, Loral did not have any concept similar to the FSV or the FMU. Hence, the screen was divided into vertical bands, and windows of a given label were confined to a single band. By dividing the screen up, the screen and the system are much less usable than in the approach described here.

Compartmented Mode Workstations (CMWs) [CMSREQS87] provide similar functionality to TX at a lower cost. Because of their lower assurance, the X server is included in the TCB, and hence is able to take advantage of existing graphics hardware. Thus, they have no need for special purpose hardware as is proposed here.

The Secure Computing Corporation is conducting research on a TCB that supports policies in which the VGS memory regions have non tranquil security attributes. With such a TCB, applications in which the required number of separate displays grows to exceed the hardware limited number of VGSs can still be supported with minimal impact on performance. The TCB makes it possible to have displays that are in a "hot backup" state which can be displayed very quickly as needed. The integrity of the separation between the different X servers that use a single VGS in sequential order is assured by the TCB's enforcement of the security labels on the VGS's memory region and the separation capabilities of the system's security policy. Control over the transition between one of the hot backups and a currently active X server is done by a very simple display controller subject that makes use of existing TCB control facilities to change the security label on a VGS. The operational view is similar to the existing multiple display X Window managers. Each display screen would have windows associated with subjects at different security levels, but different display screens could have different groups of applications. Cut and paste between windows and across screens is supported and controlled by the system security policy.

6. Conclusions

We believe that the design presented here is feasible for a low cost, yet high assurance windowing system. By leveraging the existing TX research

performed by TRW, a board could be manufactured for as little as \$300 that would allow a high assurance operating system to incorporate high performance windowing.

Such a board could be used for other purposes as well, unrelated to high assurance computing, as follows:

(1) If the screen can be reasonably partitioned, the board could be used to provide parallel video processing: multiple applications could simultaneously use the graphics hardware for high performance display. Thus, the board could be considered as a MIMD (Multiple Instruction Multiple Datastream) parallel processing graphics engine.

(2) The board could be used to provide fast switching between multiple desktops. A user might have a desktop for software development and a separate desktop for documentation writing, each of which runs different applications. Switching among the desktops does not require the applications to redraw their windows, but only requires updating the FSV.

(3) The board could be used as a development platform for new windowing systems: a user could run a windowing system for development using one VGS (and associated VFB) and use a separate VGS and VFB as a testbed without fear that a bug in the test windowing system would crash the development environment.

While (2) and (3) are feasible using the TRW TX prototype (as described in [VWS92]), (1) is only possible using the hardware solution proposed here.

7. Acronyms

BDU	Bus Decode Unit. The hardware logic for decoding bus operations and passing them to the appropriate VFB, FSV, or VGS, and for placing replies from the VFB, FSV, or VGS back on the bus.
FMU	Frame Merge Unit. The hardware logic to merge the VFBs into the PFB by selecting pixels based on corresponding values in the FSV.
FSV	Frame Selection Vector. A memory buffer used to select each pixel from the appropriate VFB.
PFB	Physical Frame Buffer. The binary image of the physical screen.
RAM	Random Access Memory.

TX/DM	Display Manager. The portion of TX responsible for managing the display and rendering window labels.
TX/IM	Input Manager. The portion of TX responsible for managing keyboard and mouse input.
TX/MS	Mini Server. The portion of TX responsible for rendering the trusted path display.
TX/SLS	Single Level Server. An untrusted X server running at a single MAC label. For every TX/SLS, there is exactly one VGS, one VFB, and one TX/WM.
TX/TSH	Trusted Shell. The portion of TX responsible for the trusted path user interface.
TX/WM	Window Manager. An untrusted X window manager running at a single MAC label. For every TX/WM, there is exactly one VGS, one VFB, and one TX/SLS.
VFB	Virtual Frame Buffer. The binary image of the screen associated with a TX/SLS or TX/MS. For every VFB, there is exactly one VGS, one TX/SLS, and one TX/WM.
VGA/SVGA	Video Graphics Array/Super Video Graphics Array. The standard for IBM PC graphics hardware.
VGS	Virtual Graphics Subsystem. A single-level graphics hardware subsystem. For every VGS, there is exactly one VFB, one TX/SLS, and one TX/WM.

8. Acknowledgments

The author appreciates the encouragement of his colleagues at Secure Computing Corporation for encouraging him to write this paper. Comments from the anonymous referees were also very helpful in improving the quality of the paper. Finally, we acknowledge the Trusted X project at TRW, which plowed the ground in which this idea grew. Key team members on that project (in addition to the author) were Hilarie Orman, John McHugh, Rita Pascale, Marty Branstad, Ann Marmor-Squires, and Doug Rothnie.

9. References

- [CMWREQS87] John P.L. Woodward, *Security Requirements for System High and Compartmented Mode Workstations*, DIA Document number DDS-2600-5502-87, November 1987.
- [Loral91] Richard Sherman, George Dinolt, and Frank Hubbard, *Multilevel Secure Workstation*, U.S. Patent 5,075,884, December 24, 1991.
- [TCSEC85] National Computer Security Center, *Trusted Computer Systems Evaluation Criteria*, DoD 5200.28-STD, Fort Meade, MD, December 1985.
- [TXArch93] *A High Assurance Window System Prototype*, J. Epstein, et al, *Journal of Computer Security*, Vol. 2, No 2&3, 1993.
- [VWS92] *Virtual Window Systems: A New Approach to Supporting Concurrent Heterogeneous Windowing Systems*, R. Pascale and J. Epstein, *Proceedings of the 1992 USENIX Summer Conference*, San Antonio TX, July 1992.

WWW Technology in the Formal Evaluation of Trusted Systems

E.J. McCauley
Trusted Systems Project Manager
Silicon Graphics Computer Systems Inc.
2011 N. Shoreline Blvd.
Mt. View, California

Abstract

The World Wide Web (WWW) introduces exciting possibilities for the use of new technology in the formal evaluation of trusted systems. This is a report of a work in progress. It discusses the conceptual foundations of the WWW use in formal evaluations of a the security properties of a system, and offers some of the initial insights gained in its use. Silicon Graphics® is using this structure for the submittal of documentation for the formal evaluation of the Trusted IRIX™/CMW 6.2 operating system.

Background

The World Wide Web is... This is an extremely difficult sentence to complete. On the purely objective level it is an extremely loose federation of independent systems connected to the Internet that offer support for the simple protocols used to retrieve information and display it. This explanation, while factually correct, trivializes the impact of this technology. The WWW empowers its users in a way that no previous technology has since the invention of the printing press. Essentially every user can create connections between concepts that are unique to the individual.

Users access the web through client programs called "browsers." These programs are available for essentially all personal computers and workstations. The browser retrieves information from a server by making a request using a simple name called a Uniform (or Universal) Resource Locator, "URL." These are the cryptic strings starting to be found at the end of advertisements. URLs describe the location, file and accessing protocol for the information. For example, `http://www.sgi.com/index.html` says how to access the information, here use the HyperText Transport Protocol, "http"; which system is to be accessed, here "sgi.com"; and the specific file or "web page", here "index.html", which also the default name if it was omitted. The information retrieved identifies its own format, so the appropriate processing by the browser may be performed to present it to the user, e.g., display text or images, play sounds or movies, or even navigate through a downloaded 3D virtual world.

One of the most powerful aspects of the WWW is the use of the Hypertext Markup Language (HTML). HTML allows the connection of documents through hypertext links. In essence, any point in one document can be connected to an arbitrary point in another document anywhere on the WWW.

In cooperation with the National Security Agency, Silicon Graphics Computer Systems Inc. (SGI) is using this technology in the submission of materials for the formal evaluation of Trusted IRIX/CMW 6.2. We feel that the use of WWW technology has the potential to significantly improve the timeliness and thoroughness of formal evaluations.

Genesis of the Concept

As the National Computer Security Center team completed the formal evaluation of the Trusted IRIX™/B 4.0.5 EPL operating system, the team at SGI universally felt "there must be a better way" to produce and submit the evaluation materials. Significant resources were expended to ensure consistency of points stated in several different documents. It was difficult to take "vertical slices" through the over 3,000 pages of submitted material to explore some specific topic from its highest level discussion down to the details of test results. The material had been developed using the "venerable" *troff* program, which required a great deal of effort to achieve the desired format.

SGI had switched to on-line electronic documentation in 1992, so there was considerable internal experience with production of documents for the IRIS Insight™ viewer. This package is based on the Standard Generalized Markup Language (SGML), and features a rich environment with embedded figures, audio, and full document indexing for rapid access by keyword searches. However, on further research, it became clear that the richness of the Insight environment came at a cost in the difficulty of creating the documents. While such efforts could be justified for customer deliverable documents, this approach appeared to be just trading one set of problems for another for evaluation submittal documents.

Fortunately, SGI had begun to embrace the WWW technology both as internal information management vehicle, and as a product technology. The first WWW products from SGI, the WebFORCE™ family of products, were released at nearly the same time as the completion of the formal evaluation of Trusted IRIX/B 4.0.5 EPL in the spring of 1994. The team began a low intensity "proof of concept" experiment of casting portions of the submittal materials into web pages. This initial experiment was extremely successful. In very short order, the team decided that our next evaluation would be documented using web pages.

As work began on the next generation system, Trusted IRIX/CMW 6.2, we elected to do all design documents as web pages. As they were completed and reviewed, they were woven into the expanding web of documentation for the system. It was also fairly easy to "recycle" the previous evaluation documents into HTML and to update them. This effort was aided by the WebMagic™ editor, a screen oriented editor for HTML. One of the helpful features of WebMagic is the ability to establish hypertext links within and between documents with a simple point and click interface. HTML documents could also be edited with conventional text editors, and could be managed through our standard configuration management tools.

A system evaluation is a cooperative effort between the vendor and the team assembled by NSA/NCSC. If the documentation were to be submitted as web pages, it would be necessary to seek NSA approval for this form of submittal. It is difficult to appreciate Web technology in the abstract, so the SGI team decided that the most effective way to present the concept would be to relate it to the guidelines for submittals produced by the Process Action Team Guidance Working Group, which had established guidelines for the submittal of information to NSA/NCSC for formal evaluation of systems. In August 1995, SGI presented a demonstration of the technology for NSA/NCSC. This demonstration showed an HTML version of the "PAT Working Group Form and Content of Vendor Design Documentation" report, which linked into the top level description of Trusted IRIX/B 4.0.5 EPL found in the Final Evaluation Report, which had also been converted to HTML. The sections of the Final Evaluation Report were linked into the major design documents for the system. Additional linkage structures tied in the system manual pages (primary interface documentation), test plans and test results. The overall structure is shown in Figure 1.

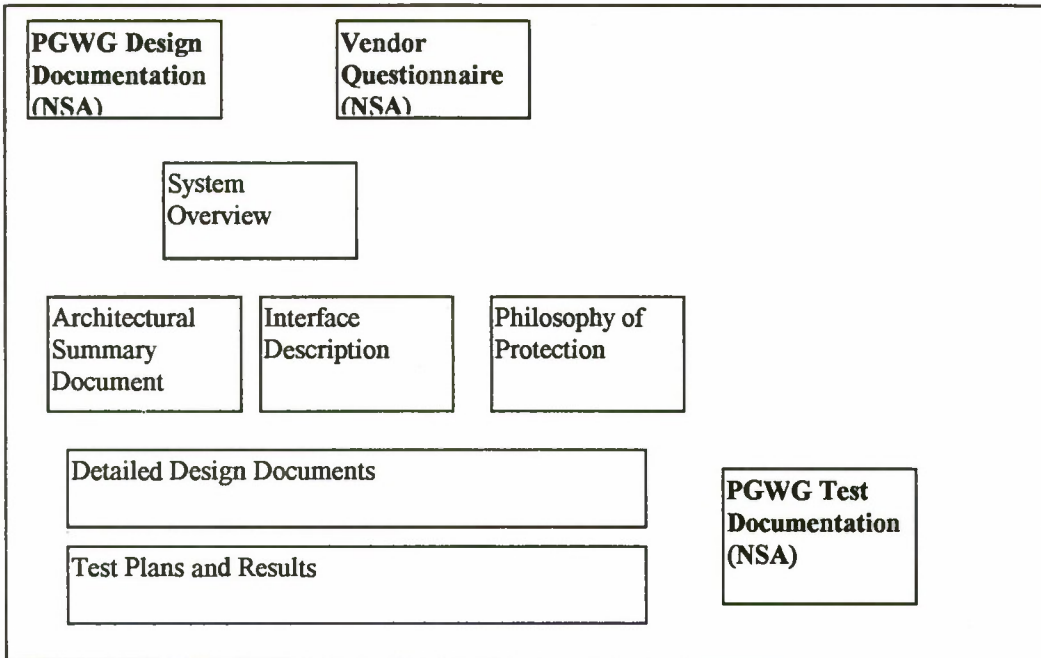


Figure 1 -Overall Web Structure

The demonstration generated considerable interest, and by the time of the 1995 National Information Security Conference, NSA had convened an informal working group, dubbed the "Hypertext Working Group," to provide more details and guidelines for the use of WWW technologies in the submittal of formal evaluation documentation. This team submitted its report to NSA in February 1996. At the outset, the report urged flexibility to make maximum use of emerging technology. The report suggested starting with the two PGWG reports, and the NCSC Vendor Questionnaire. This illustrates a strength of the WWW technology. These three documents point to the same underlying information in the submittal. Depending on the nature of the task at hand, the evaluation team can access the information through several different paths. This principle can be extended by creating other access paths structured to needs of the evaluation team. Most browsers also support more informal indexing through their ability to create "bookmarks," URLs that are used to "remember" interesting content and rapidly return to it.

Browsers also facilitate information access by remembering the path used to reach some URL, and being able to backtrack to earlier points in the search. This allows an evaluator to research a specific detail and to go back to where the search started to look at other information.

One area that received much attention and discussion by the Hypertext Working Group was the mechanisms for coordination and feedback between the evaluation team and the vendor. These efforts are collaborative, and there is a need to generate feedback to the vendor, for the vendor to respond by altering the system software and documentation, and for the team to be able to assess the results looking at "before and after" content. Several different mechanisms were discussed, and recommendations were made to provide links to "before" content in the "after" document. More sophisticated schemes involving on the fly generation of the differences were also considered. Support for annotation of documents is a very active development area for the Web, so new technologies may overtake these recommendations.

The use of WWW technology changes the environment as compared to conventional document submittals. The most significant of these changes is that HTML documents lack a page structure. This means that conventional schemes like indices and tables of contents must be replaced by hypertext links. The lack of a page structure also posed challenges for the appropriate marking of vendor confidential materials. Many evaluation submittals contain information that the vendor considers sensitive, which must be clearly

indicated to the members of the evaluation team. SGI is exploring several ways to do this; one approach has been to have a background that indicates the sensitivity of the document, as well as specific sensitivity markings at the beginning and end of the document.

An unexpected issue arose as the working group discussed the mechanics of the evaluation. Many evaluators work in environments with poor or non-existent connectivity to the external Internet, due to the security considerations of their environments. To address this, the working group recommended that the evaluation submittal be self contained and not contain links to sites on the Internet. This is somewhat unfortunate for the SGI submittal, as much of the material on the processors and systems is available as web pages on the SGI and MIPS® Technology Inc. corporate web pages. The submittal package has made copies of this material for use by the evaluation team.

Conclusions

Since the evaluation of Trusted IRIX/CMW 6.2 is not complete, it is a little premature to draw too many conclusions. Still, a number of things have emerged from the initial work. Most important is that the existing framework for evaluation submittals fits well to the new technology of the web. It has been straightforward to take documents developed for conventional submittals and adapt their structure to a web based submittal. We have found that creation and management of web based documents is actually easier than the techniques used in the earlier evaluation, especially since there is very active tool development to aid the task.

Acknowledgements

This work has been a collaboration of many peoples' efforts. In addition to the SGI Trusted IRIX™ team, we appreciate the support of Dennis Kinch, Janine Pedersen and Rita Montequin of NSA in seeing the potential of the work and establishing the Hypertext Working Group. The working group, Jim Reynolds (MITRE), Richard Waltzer (MITRE), Al Nims (Aerospace), Jack Walsh (NSA), and Casey Schauler (SGI) all contributed insights and observations that materially improved the initial ideas.

Trademarks

Silicon Graphics and MIPS are registered trademarks, and IRIX, IRIS Insight, and WebFORCE are trademarks of Silicon Graphics, Inc.

THE CERTIFICATION OF THE INTERIM KEY ESCROW SYSTEM

Ellen Flahavin & Ray Snouffer
National Institute of Standards and Technology
Building 820, Room 414
Gaithersburg, MD 20899
(301) 975-3871 & (301) 975-4436
flahavin@csmes.ncsl.nist.gov
ray.snouffer@nist.gov

1. INTRODUCTION

The U.S. Government Key Escrow System (KES) provides for lawfully authorized access to the key required to decipher communications secured with products built in conformance with the Escrowed Encryption Standard, Federal Information Processing Standards Publication (FIPS) 185. This paper is intended for presentation at the 1996 National Information Systems Security Conference. The objective of this paper is to describe the certification and accreditation of the Interim KES and provide a historical overview of the Key Escrow Certification Working Group's (KECWG) activities. The defined purpose of the certification working group is to perform a certification on both the interim and the final KES in accordance with the Guideline for Computer Security Certification and Accreditation (FIPS 102). FIPS 102 provides guidelines for computer security certification and accreditation of sensitive computer security applications. The National Institute of Standards and Technology (NIST) chairs the KECWG. In addition to NIST, the membership consists of the Department of Justice (DOJ), the Department of Treasury, the Federal Bureau of Investigation (FBI), the National Security Agency (NSA) and the Department of Commerce (DOC).

1.1 The National Key Escrow Program

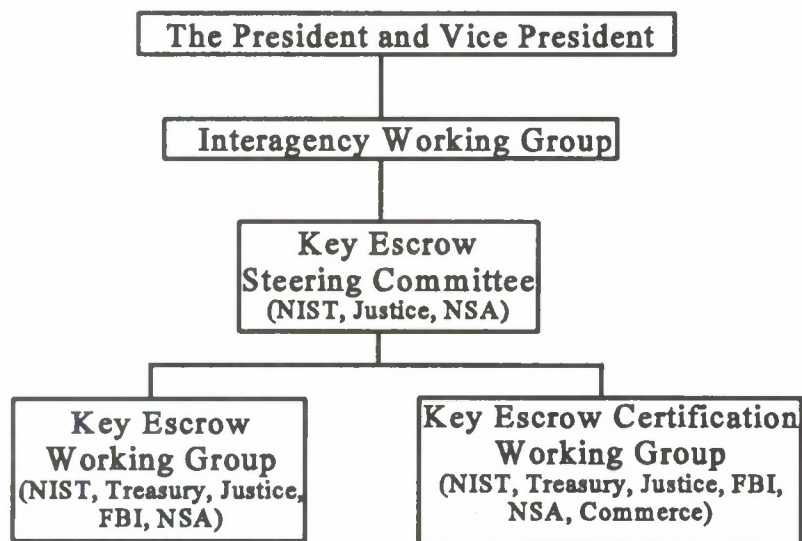
The Key Escrow System was developed to support of the U.S. Government's Escrow Encryption Standard (FIPS-185) and Presidential Decision Directive/NSC-5. The primary objective of this program is to provide the U.S. Government and the private sector with high-quality, secure communications products without jeopardizing effective law enforcement, public safety, and national security. The initiative is based on a special tamper-resistant hardware encryption device (Clipper/Capstone Chip) and a KES. The KES is an interagency program, with support from NIST, DOJ, Treasury, FBI, and NSA. NIST serves as the National Program Manager for Key Escrow; overseeing the current Interim System and the development of the Final System.

1.2 Roles, Responsibilities and Organization of the Participating Agencies

Each agency participating in the KES provides unique support to the program. NIST serves two distinct roles, that of Program Manager and one of the two Escrow Agents. Treasury serves as the second Escrow Agent. The Department of Justice acts as the System Security Manager, system accreditor, and one of the two Family Key Agents. The FBI serves as the second Family

Key Agent and as the initial Law Enforcement Agent. The National Security Agency is responsible for system development and system engineering support.

The overall program is supported through a system of working groups and committees. The Vice President of the United States heads the organizational structure. The Inter-agency Working Group (IWG) provides senior level support to the Vice President on technology and policy. The KES Steering Committee was established to provide the participating agencies with a senior level forum for discussing and resolving issues arising from the interagency nature of the program. Overall KES policy and budget issues are also the responsibilities of the Steering Committee. Further support for the program is provided by the Key Escrow System Working Group (KESWG) and the KECWG. Additional information on the KESWG and the KECWG and their role in system certification are provide in sections 1.4 and 1.5. The following diagram shows the structure of the KES Program.



1.3 Basis For Establishment of the KES Certification Working Group

The KES helps to ensure that the unique keys and key components are released only for legally authorized surveillance activities and only for the duration of the authorization period. At its June 3, 1994 meeting, the KES Steering Committee agreed that the KES would be certified by a committee consisting of representatives from appropriate government agencies and accredited by DOJ. The National Program Manager for Key Escrow established the KECWG to certify both the interim and the final KES. The KES certification will be used as input to the corresponding DOJ accreditation.

1.4 Scope of the C&A Effort

Certification is required by Circular A-130 of the Office of Management and Budget (OMB), for

all computer applications processing Sensitive Unclassified (hereafter referred to as Sensitive) information. FIPS PUB 102, Guideline to Computer Security Certification and Accreditation, 27 September 1983, and the NSA Draft C&A Process Handbook were used to define the certification methodology employed. FIPS PUB 102 presents, in detail, an approach to developing a certification and accreditation program. The NSA Draft C&A Handbook provides a technical process for certifying applications. The activities of the KECWG include the following:

- Writing the Certification, Security, and System Test and Evaluation Plans,
- Implementing the certification process,
- Other tasks specified in FIPS 102 that the KECWG believes necessary for certification,
- Evaluating the Risk Assessment and developing a Statement of Residual Risk, and
- Providing a recommendation and a certification package to the KES Accreditor.

1.5 The Key Escrow System Working Group

The KESWG was established concurrently with the KECWG by the National Program Manager for Key Escrow. However, The KECWG is an independent group and does not receive direction from the KESWG. The purpose of the KESWG is to manage the development and operation of the KES under the guidance of the Key Escrow Steering Committee. The National Program Manager for Key Escrow reports the activities of the KESWG to the Steering Committee. The activities of the KESWG include the following:

- Developing the KES, including subsystems and documentation,
- Operating the KES,
- Baselining documentation for hardware, software, and operational procedures,
- Establishing and maintaining the KES Configuration Management Process,
- Planning and implementing improvements to the KES, and
- Establishing operational agreements between its members.

Additionally, the KESWG is responsible for developing documents which are essential to system certification. These include the KES Security Policy and the KES Protocols and Procedures (P&P). All system certification testing for the Interim KES is based on the KES P&P document. The KES Security Policy serves as the basis for the KES Security Plan.

2.0 SECURITY ENGINEERING

Security engineering (including C&A) for KES was included as part of the system engineering life cycle. The security engineering tasks were developed and executed concurrently with the system design and development activities. These activities included: developing a security architecture, defining security requirements, and preparing a System Security Plan (SSP).

2.1 Define Security Requirements

The KESWG along with the NSA system developer agreed on a set of security requirements and

identified implementation issues. To ensure compliance with the requirements, reviews were held between the KESWG and the NSA system developer. The system developer is responsible for defining requirements, developing the KES architecture (specifications), as well as designing, implementing and testing the KES. The KECWG is also responsible for testing the system. At the completion of the requirements definition process, the security requirements were included with the functional requirements in the KES Security Policy. This document was reviewed and approved by the KESWG.

2.2 The C&A Process

Certification of the KES involved a technical assessment of the security functions to determine the extent that these functions met the KES security requirements and the KES Security Policy. This certification also included executing security tests to demonstrate the adequacy of the security features and requirements. The test results were included in the certification package for review by the system accreditor. Accreditation is a management decision by DOJ which is required prior to declaring the KES operational.

FIPS PUB 102, Guideline for Computer Security Certification and Accreditation, was used to develop the KES C&A Process. This standard was used because the KES is authorized and staffed by federal agencies; and is used for the processing of sensitive information. NIST is responsible for providing guidance to agencies that process sensitive information. As defined in FIPS PUB 102, the certification effort is divided into basic and detailed evaluations. Detailed evaluation focuses on whether or not specific security features operate correctly.

The NSA draft C&A Handbook was used to further define the KES C&A process. The handbook was used by the KECWG to tailor the certification efforts to the particular purpose, environment, degrees of assurance, and criticality of the system as well as threats to the system.

Phase One C&A focused on existing physical and administrative/personnel security because of the limited number of implemented technical security features. The goals of this phase were to test the KES Protocols and Procedures (P&P), evaluate the certification process itself, and make refinements to the process prior to beginning Phase Two C&A. Since the interim system is primarily manual, the testing was performed sequentially.

2.3 Functional Certification Tasks

Step 1 - Identify the System

The purpose of this step was to identify system specific information that would impact the certification effort. This information included identifying the Accreditor, committing resources by management, establishing the system boundary, and determining the certification type. The determination of the certification type included looking at key aspects of the system such as assurance, confidentiality, integrity, authenticity, and availability. This determination indicated that a type 3 moderate certification as specified in the C&A Handbook, was required. This type of certification is more detailed and complex and is generally used for systems that require higher

degrees of assurance, have a greater level of risk, and are more complex.

Step 2 - Planning

The second step was to develop a certification plan. This plan consisted of determining the composition of the certification team, incorporating milestones, obtaining necessary resources, and documenting planning information.

Step 3 - Perform System Analysis

A comprehensive analysis of both the technical and non-technical security features and other safeguards of the system was performed. The first activity performed involved analysis of the detailed system documentation to determine if and how the security requirements were met. When necessary, additional documentation was developed. Other major activities included performing system testing (see Section 3) and conducting a risk analysis. The analysis established the extent that the KES met the security requirements defined.

A risk analysis group was formed by NSA to assess the appropriateness of the safeguards to minimize risk, while the security testing focused on the functionality and effectiveness of the safeguards.

Step 4 - Report Findings and Recommendations

This step involved documenting and coordinating the results of Step 3, and preparing a recommendation and certification package. The certification package contains a set of supporting documentation including: test results, risk assessment, and the KES P&P. The KECWG also provided a recommendation and statement of residual risk to the Accreditor. The purpose of this total package was to assist the Accreditor in approving the system for operation.

3.0 System Testing

The testing of the Interim KES was a required step in the C&A of the system. DOJ served as the system accreditor for the Interim KES and was charged with ensuring that the system was adequately tested prior to accreditation. The KECWG was formed to assist DOJ with the accreditation of the system by developing test plans, providing organization for the tests, and serving as the test coordinator.

3.1 Test Organization

The system tests were divided into two phases: a walk through of KES P&P and the official test. The walk through differed from the official test in several important aspects. The walk through followed the P&P to ensure that all procedures for each subsystem were adequately documented. However, the procedures were not necessarily performed sequentially. During the walk through,

all tests involving the extraction and release of keys utilized test keying materials. Also, no oversight by the system accreditor was required for the walk through. The C&A contractor was present during the walk through to provide organization and to note corrections to the P&P and the draft test plan. An additional benefit of the walk through was to familiarize the staff with the various pieces of KES equipment and procedures prior to the official test. The walk through took place during May 1995.

The official test was divided into the testing of each of the KES subsystems following the documented procedures of the P&P. The test was not considered "end to end" because no single chip was not tracked throughout the entire process (programming through release), and the tests did not necessarily follow a sequential format. The testing of the extraction and release of encrypted Key Components (KC) did follow a chronological format and was conducted "end to end" during the course of one day. The tests were observed by the system accreditor (or representative) and one independent observer at each subsystem site. The testing of the programming site occurred during the June 5, 1995 programming session at Mykotronx in Torrance, CA. The official test of the encrypted KC release was performed on November 4, 1995.

All results and observations were reviewed by the KECWG and included in the accreditation package, which was sent to DOJ. Recommendations for changing procedures resulting from the walk through and the official test were handled through the KES configuration management process. Though the official test was a comprehensive "positive test" utilizing the best case scenario, all procedures were thoroughly tested and all agencies participated.

3.2 Programming Site Testing

Certification testing was conducted during a regular chip programming session at the programming site in Torrance CA. Chip programming was conducted over a period of one week. All testing associated with programming was supervised by a certifying official and an independent observer. The test was positive, with no errors intentionally inserted. However, participants were asked to document appropriate areas for future negative testing. Testing included physical security procedures, programming device initialization, key component generation, chip programming, software archiving, system sanitization, and key component transportation. Test logs were based on the KES P&P, and followed a chronological format. Modifications to the P&P were noted for inclusion in the next release of the P&P document.

Preparatory work for the programming session was also tested and documented at each Escrow Site. This included physical security, computer initialization, development of programming seed materials, and transportation of the seed materials to the programming site. Upon return to the Escrow Sites, the key component storage procedures were also tested and documented.

3.3 Extraction and Release Testing

Certification testing of the key component release occurred during regular working hours at NIST, Treasury, DOJ, and the FBI. The test was conducted during the course of a single day.

Actual extractions and releases may be required during off hours; however this test was set up to simulate the most ideal scenario. Timing information was collected during the official test. This information was not used to accredit the timing for the extraction and release, but was used to estimate the time required for each process. No errors were intentionally introduced during this test, but the teams of evaluators and test participants were instructed to note areas where appropriate error testing could be incorporated. The test coordinator maintained contact with each of the teams by phone during the test and attended the test of the decryption process. In order for the test to be conducted simultaneously at all subsystem sites, three teams of observers were formed. Each team had two members: one representing the system accreditor and one serving as independent observer.

The test of the extraction process utilized key components for AT&T Telephone Security Device Model 3600 (Clipper Chip based phones) owned and retained by the FBI. The Authorization for the release of key components for these devices for testing purposes, was granted by the United States Attorney General on August 23, 1995, in a memorandum to the Director of the FBI, Louis Freeh. The authority for the intercept was granted for a period of one month and was discontinued upon the completion of testing. The discontinuation of the intercept prior to the end of the deadline is consistent with the guidelines set forth by the United States' Attorney General.

The FBI provided both the facility and equipment for the intercept and decryption of the communications. The process utilized the internal phone system at the FBI facility and two FBI owned AT&T TSD3600s. A representative from DOJ, one independent observer, two Escrow Officers, and two FBI agents were required during the test of the decryption process. The Escrow Officers providing the extraction diskettes witnessed the decryption process; though this would not be allowed during an actual intercept situation.

4.0 SUMMARY

In order to be successful, a multi-agency certification of a National system requires coordination, planning, and established structure. The following actions are essential to the success of the C&A effort.

1. Establishing a charter enabled the KECWG to define the purpose of the group, establish the group's organization, outline required activities, and define the decision making process.
2. Ensuring that the KECWG members from the federal agencies were fully authorized to represent their agency, allowed critical decision making during the meetings.
3. Defining the security requirements early in the system life cycle provided a solid basis for testing, and ensured system security at each test point.
4. Developing the residual risk statement in a group setting provided the membership and

the accrediting authority with a full understanding of the risk analysis.

5. Having the accrediting authority actively involved from the beginning of the certification process provided an assurance of final accreditation.

There were two areas that were not as effective and should have been performed differently. First, the risk analysis was performed independently by an outside group without shadowing by the certification working group. Since the analysis was accomplished without the participation of the KECWG, modifications to the system and procedures could not be dynamically introduced into the process. Thus, the analysis required an update upon its first review by the KECWG. Second, the chair of the KECWG should have had more authority to enforce deadlines and scheduling. The enforcement of deadlines was made more complex by the interagency nature of the project.

At the writing of this paper, all certification activities have been completed and the certification package is being compiled. Once complete, the certification package will be sent to DOJ for approval and system accreditation. The completed activities have been very successful and the certification process has yielded several unexpected benefits. Certification testing identified areas in the P&P where additional granularity was required. It also pointed out areas where procedures could be optimized. The test results and comments from the testers were folded into the current P&P baseline. In addition, the System Test and Evaluation Plan will form the basis for the overall KES Test Plan. The KES Test Plan will be used for both system testing and training for the Escrow Officers. The development of the statement of residual risk provided an open forum for the discussion of both the physical and technical security of the system. These discussions provided the KECWG with additional assurance of the security of the system.

REFERENCES

National Institute of Standards and Technology, "Federal Information Processing Standards Publication 185, Escrowed Encryption Standard", 4 February 1994.

National Institute of Standards and Technology, "Federal Information Processing Standards Publication 102, Guideline for Computer Security Certification and Accreditation", 27 September 1983.

National Security Agency, "Certification and Accreditation Process Handbook (Draft)", NCSC-TG-031, February 1994.

Office of Management and Budget, "Circular No. A-130, Management of Federal Information Resources", 12 December, 1985.

Configuration Management
in
Security related
Software Engineering Processes

Klaus Keus, Thomas Gast *
Bundesamt für Sicherheit in der Informationstechnik
Postfach 20 03 63, D - 53133 Bonn

* e-Mail: {gast, keus}@bsi.de

Abstract

IT-Security requires specific enhancements and tailoring during the complete life cycle of the product or system, including a security focused SW-engineering process. One of the key technologies to manage the software engineering process is the use of a tool driven Software Configuration Management.

Software Configuration Management (SCM)[10] is an aspect of establishing that the functional requirements and specifications are realised in the implementation during the whole life cycle. SCM is the activity of controlling the software product by managing the versions of all components and their relationships. It is one of the fundamental activities of software engineering in general and becomes most important in the development of high assurance software as in IT-Security.

This paper demonstrates, that the management of the whole software life cycle using SCM guarantees the traceability from the requirements specification (phase) via the design and the implementation phases to the final software product and maintenance phases by coordinating/controlling the changes in all phases of the software engineering process.

Using SCM with defined roles and access control enables the implementation of security measures to manage the software engineering process in a defined and controlled way. Thereby the assurance of the development process itself will be improved.

This paper discusses basic requirements of a Software Configuration Management System to suit the field of IT-Security. The scope of these requirements extends from quality standards, such as the ISO9000, to the specifics in general accepted "IT-Security Evaluation Criteria", such as the ITSEC (Information Technology Security Evaluation Criteria)[6] and the CC (Common Criteria)[2]. A first approach to a maturity model for SCM in IT-Security will be given.

1. Introduction

Configuration Management delivers the key to transparency in the software engineering process. It is a precondition for successful manufacturing of high quality software products. Configuration Management becomes more and more important with increasing complexity of the software products and the software engineering process itself. Without a defined change and integration management there is no way to control of the software engineering process. Software Configuration Management (SCM) increases the quality and the assurance of the software engineering process with direct impact on the quality and assurance of the IT-products. How to run a SCM in the special environment of the development of critical software systems in general, and of high security software systems in special, will be discussed in the following chapters. Focus of the paper are the more technical aspects. Although the organisational and the management issues are very important and have to be respected in a more global and common view, these issues are not respected in this paper.

Chapter 2 will briefly introduce the general scope and functionality of Software Configuration Management. The terminology will be defined. It will be discussed what SCM really is, how it works and why we need it.

In chapter 3, general requirements for a SCM will be presented. The requirements will be derived from the quality standard ISO 9000 part 3 [8]. It deals with general models and criteria like the V-Model [13][VMOD] from the German KBSt, the Capability Maturity Model (CMM)[3] and the ESA Software Engineering Criteria [5]. These will be considered directly or indirectly. These requirements will be mapped to basic security requirements derived from the Information Technology Security Evaluation Criteria (ITSEC). Ways to fulfill these conditions and requirements will be discussed. A survey of the impact to the quality and assurance of the software engineering process and the software product will be given.

In chapter 4, special security requirements to a SCM will be presented. The requirements will be derived from the Information Technology Security Evaluation Criteria (ITSEC), and consideration will be given to the Common Criteria (CC), the Canadian Trusted Computer Product Evaluation Criteria (CTCPEC)[4] and the Security Engineering CMM. Possible ways to comply with these requirements will be offered. An analysis and survey of the impact on the assurance of the software engineering process and the software product will be presented.

In chapter 5 the state of the art in SCM will be explained. A SCM maturity level model will be introduced. The application of technical solutions using SCM tools will be discussed.

Chapter 6 will present some future aspects concerning the integration of SCM in the development process of high security software products. Some control approaches will be discussed to ensure the required impact of the actually installed SCM on the assurance of the software engineering process and the resulting software products, stretching from process and product evaluation to the accreditation of the entire software manufacturing process.

2. Software Configuration Management

The most frustrating software problems are well known by every software professional in the form of the reappearance of recently fixed bugs, mysterious disappearance of implemented and tested features, or even the total failure of a fully tested program. In our view such problems may lead to severe security problems if they are not discovered before shipment and installation. Therefore first we have to investigate where these problems derive from.

The increasing complexity of the software products shown in figure 2.1 has direct impact on the necessary coordination of the software manufacturing process. Many different people of possibly different institutions and/or locations may work on the same project. It is not possible and sufficient to build a rigid integration plan because the coordination of the parallel development activities must cover the complete software life cycle.

The coordination of designing such software products has to solve conflicts resulting from problems such as simultaneous update, shared code, common code, or multiple versions. The key is the qualified implementation and maintenance of a control system.

SCM reflects the current configuration and status of the software at any time, controls any changes to any object of the software, and maintains the traceability throughout the whole software life cycle. Also special security aspects, e.g. integrity, confidentiality and availability, are issues in the scope of SCM. SCM ensures the integrity and consistency of the software throughout the manufacturing process. SCM delegates the responsibility to check out the software modules to the developers, testers and so on in a defined way, to guarantee special confidentiality requirements on the "Need-to-Know"-principle in the manufacturing process of high security software components. SCM ensures the availability of any software configuration at any time.

SCM is a complex task that itself has to be driven in a well defined manner as shown by figure 2.1.

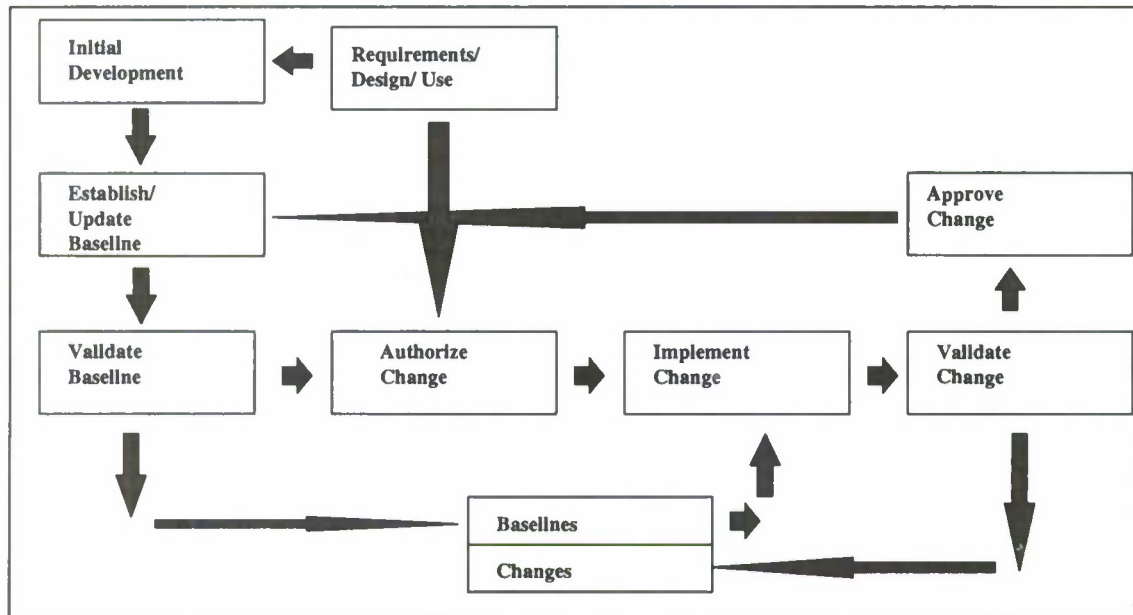


Figure 2.1 Well Defined SCM Process

37

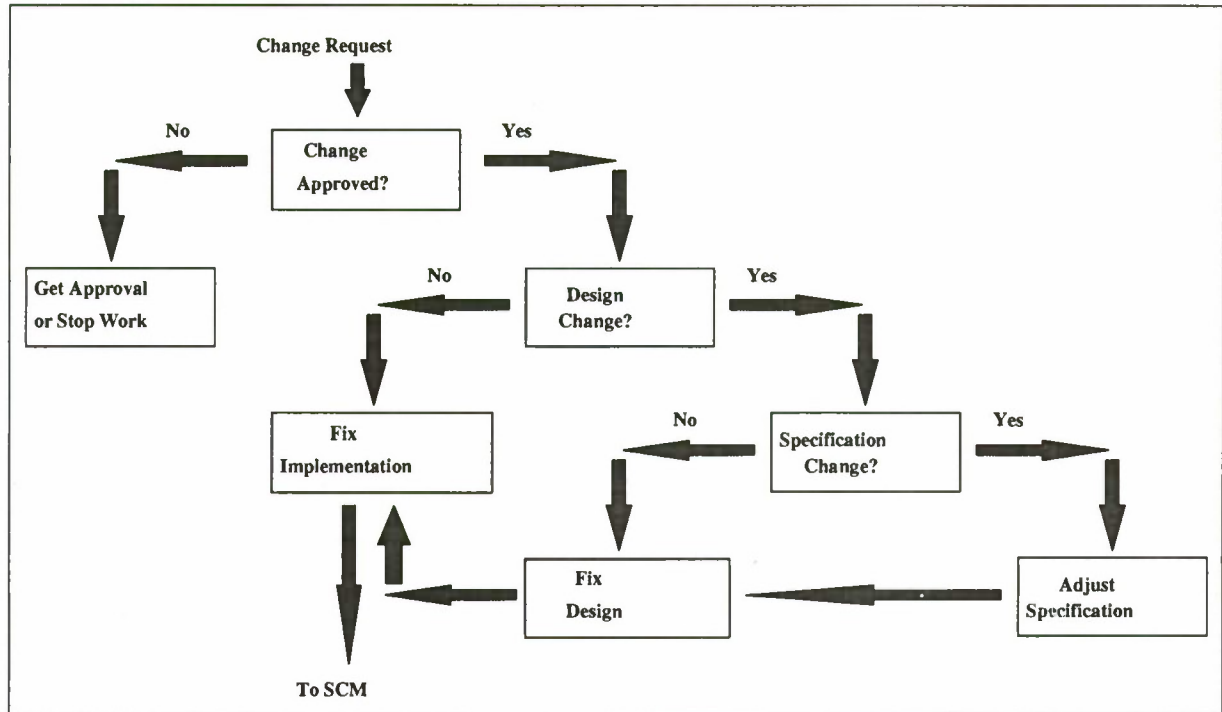


Figure 2.2 Well Defined Implementation Change Control Procedure

In the following chapters the requirements to the SCM will be refined to a more specific level with references to the quality and security assurance standards and criteria.

3. General Quality and Assurance Aspects of SCM

Quality aspects of the software manufacturing process in general are respected in the ISO 9000 Part 3 [8]. One of the main parts of this standard is chapter 6 dealing with configuration management. In the development of high security products it is important to analyze the impact of these requirements to the assurance of the manufacturing process and to the assurance of the software products themselves. Therefore the requirements of the ISO 9000 are taken in relation to the ITSEC if possible. The technical and administrative requirements concerning SCM are listed and the implementation of a tool based SCM to fulfill these requirements is discussed.

The first requirement to be discussed is to **identify uniquely the versions of each software item**. There is a link to the ITSEC requirement **The TOE (Target of Evaluation), its basic components and all documents provided ... shall possess a unique identifier**. It is a basic feature of SCM to guarantee an unambiguous access to any managed Configuration Item (CI) at any time. This is a precondition to control the development process as well as the process of any high security software product. These two requirements are completed by **identify the versions of each software item which together constitute a specific version of a complete product [ISO] and the configuration list provided shall enumerate all basic components out of which the TOE is built [ITSEC]**. So SCM gives evidence at any time that in fact just the specified high security product is delivered and installed. These requirements for instance may be complied by the use of unique triple identifiers in the form **name:version:type**. SCM enforces the uniqueness of the identifier all over the complete project directory tree. To avoid any confusion it should be prevented to use different PATH directives for redundant identifiers in the

project at any time. Hence any change of a CI leads to a new identifier of the unique version in the history of the CI.

SCM delivers transparency of the current state of the development process. It must **identify the build status of software products in development or delivered and installed [ISO]**. Similar to the requirements of ISO 9000 e.g. the ESA proclaims the three hierarchical state levels **Development, Controlled** and **Static** which reflect the main stations of the software life cycle. In the development process of security critical software products the granularity of the development status must be increased to the various roles in the development process in a balanced way. A mapping of the responsibilities to the current status should be possible at any time. The development status consists of 4 detail mappings: the working status maps to the activity of code development, the integrate status maps to integration testing, the SQA (Software Quality Assurance) status to the system level testing, and the released status as output status to the development process. This SCM, defined by role and status, fits best the IT security requirements for confidentiality, integrity, and availability, and leads to a high quality and high assurance software development process. SCM enforces each CI to respect this state machine. A Secure SCM (SSCM) manages the access to any CI in accordance to this development model.

An important aspect of SCM is the coordination of the various developing and changing activities of the components in the software product during the software engineering process. During the development process a complex network of dependabilities of the several modules and the objects will be built by the use of IMPORT/EXPORT features or via inheritance and granting. This network reflects the general access to data structures and functionality. To manage the consistency of this network during the whole life cycle SCM accepts a change not before its validity is confirmed and the effects to other CIs are identified and examined. This is an important requirement derived from the ISO as well as from the ITSEC to guarantee the integrity of the software product during any change process.

A most critical task of SCM is to **control simultaneous updating of a given software item by more than one person [ISO]**. Two kinds of simultaneous updating have to be distinguished: the updating of two independent versions of the CI resulting in two independent versions of the whole software product and the updating of the two independent change processes resulting in one version of the CI using controlled merging.

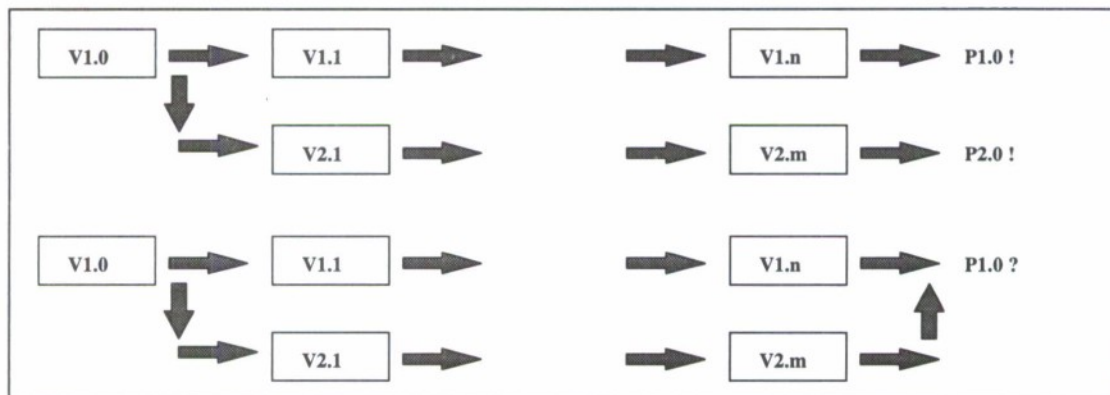


Figure 3.1 Branching; Branching and Merging

In the first case SCM has to guarantee the independence of the new branch throughout the whole life cycle of the two or more arising product versions. No merging is granted at any time.

SCM provides a clear distinction between the version identifiers of the two branches like 1.n and 2.m shown in figure 3.1. SSCM is responsible for meeting the confidentiality aspects of those independent development activities.

The simultaneous updating of a CI - intending to merge two or more checked out versions to one version in the history tree of the CI - involves some special problems to SCM. How can SCM guarantee the correspondence of the simultaneous developer's activities to avoid the occurrence of conflicts? How can SCM trace the history of the CI following a complex branch and merge tree when problems occur? How can SCM manage and control the different responsibilities of several developers who are working simultaneously on the same version of the same CI? Taking this approach all requirements for confidentiality, integrity and availability to a high assurance software engineering process will be compromised. A successive and complex branch and merge process is not manageable. This way of simultaneous updating must be rejected even at check out time and must be verified at check in. Otherwise the software engineering process will run out of control.

The control of any change process is the precondition for integrity and availability of any version of any CI. SCM controls the update process in the time frame from check out to check in to get the complete transparency of any change activity. SCM must **identify and track all actions and changes resulting from a change request, from initiation through to release**[ISO]. The configuration control tools shall be able to control and audit changes between different versions of objects subject to configuration control[ITSEC]. All modifications of these objects shall be audited with originator, date and time[ITSEC]. To fulfill these requirements a complete audit trail of any change must exist. This audit trail delivers SCM with the information of the event, the cause and the initiator of the change process. The implementation of any change request becomes identifiable and verifiable. The responsibilities of all steps in the change process are defined and documented. Following these requirements the history of any change is available to SCM.

As discussed in section 2 and shown by figure 2.1 the SCM process itself runs in a well defined manner. A configuration management plan (CMP) fixes all necessary administrative activities. The CMP defines **procedures to identify, document, review and authorize any changes to the software items** ...[ISO]. Following the ISO and the ITSEC requirements the CMP defines the responsibilities, the actions to be performed, the tools, techniques and methodologies, and it defines the baseline to check the CIs first under SCM. The CMP is part of the security policy of any high assurance development process.

SCM is applied in the whole life cycle of the software product. SCM must **maintain procedures for identifying software items during all phases, starting from specification through development, replication and delivery**[ISO]. All objects created during the development process ... shall be subject to configuration control[ITSEC].

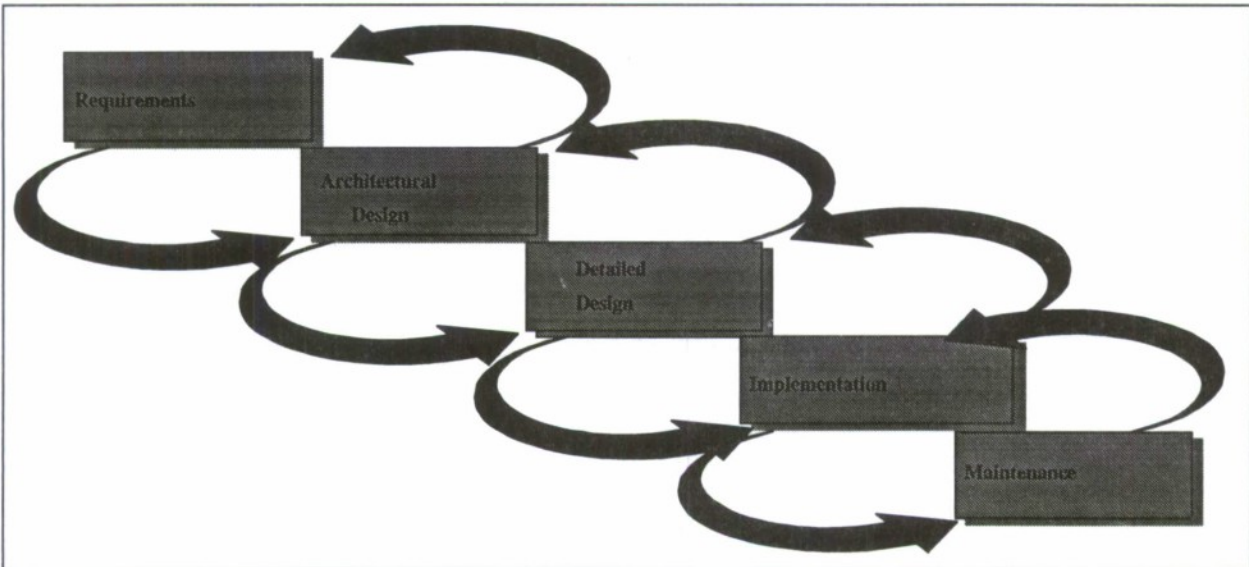


Figure 3.2 Traceability of the software engineering process

The traceability of the whole software life cycle is a precondition for the correctness of the software manufacturing process. SCM should manage all objects created during this life cycle. SCM identifies the correlations of the several versions of the specification documents following the object versions from the design, the implementation and the testing phase to the delivered version of the software product. SCM leads the development and maintenance process version driven through all phases of the software engineering process. The correlation of the several versions of the design documents of the product delivered to the according versions of the user manuals is dealt by SCM to guarantee the consistency of all product documents. This traceability aspect is the precondition of a successful security ITSEC evaluation of any software product. The scope of SCM includes more than the created objects during the engineering process. Even **all tools used in the development process shall be subject to configuration control[ITSEC]** to guarantee the rebuilding of any shipped version at any time. This includes the management of the applied CASE tools, compilers, debuggers, GUIs and operating systems. Also all external interfaces to the hardware or to software such as libraries must be under control of SCM. Following all these requirements SCM is able to guarantee the integrity and availability of any released version of the software product.

A SCM running in a defined way as shown in figure 2.1 fulfills the ISO quality requirements as discussed above and builds up a good base for the management of all objects in a high quality software engineering process. As shown in this chapter there is a close connection between the ISO quality and the ITSEC security requirements. Recognizing this aspect it becomes possible to install and to drive such a tool driven SCM which fulfills both aspects: the ISO quality and the basic ITSEC security requirements. So by the way we get a high quality SCM including basic features of a high assurance software engineering process and respecting the IT security aspects confidentiality, integrity and availability.

4. General Security Aspects of SCM

In this chapter specific security aspects of SCM are discussed. Security aspects that exceed the requirements from chapter 3 concerning the correctness of the software engineering process.

SCM in the special scope of IT security deals with problems occurring by accidental or deliberate unauthorized modifications of software components.

Security in IT deals with the access of subjects (persons or processes) to objects (data, programs, IT in general) focusing to determined existing risks. These risks are called threats. To become independent from the technical development environment, e.g. from the operating system, special security functions are required and implemented in the SSCM itself. So SSCM provides a complete set of security functionality, at least it should include functions for identification and authentication, access control, accountability, audit and accuracy.

An independent identification and authentication is the precondition for administrating the different users of a role driven SSCM. SSCM administrates roles like developer, integration manager, SQA manager, build manager and system administrator. The developer is responsible for the development of the single modules. The CI is explicitly checked out to the private working area of the developer. The access to this CI by any other developer has to be prevented. The developer drives previous tests in his separate development area (so-called α -tests, which primarily respect the functionality and the correctness aspects). The integration manager builds the final complete product by composing the several modules checked in by the developers in a step-by-step way. He is responsible for the integration testing and for the process of the construction of the current version of the product. The SQA manager is responsible for the final acceptance of the installed product version, including the responsibility for the final testing phase (so-called β -tests, including the quality tests for the product (functionality and correctness tests)). The current product version is checked to RELEASED. The build manager gets access to all released product versions and is responsible for the rebuilding of any released product version. Those CIs which have to be changed, will be checked out by the build manager. He initiates any change process by checking them out into the working area of the developers. The system administrator is responsible for the complete administration of the SSCM. This task includes the administration of the resources of the SSCM, the archiving of the CIs, the security aspects as e.g. user and access management and the maintenance of the SSCM system itself.

Based on this user administration the SSCM access control enforces a defined change process according to the state machine discussed in chapter 3. SSCM ensures **that only authorised changes by authorised persons are possible**[ITSEC]. In the development process of critical IT-products - e.g. as high security software systems - the access to security kernel modules must be restricted to a group of trustworthy developers. This approach supports a countermeasure for possible confidentiality conflicts. This confidentiality aspect has to be guaranteed throughout all successive development activities respecting the increasing granularity concerning confidentiality requirements. Meeting these requirements it is necessary to have a detailed mapping of the development activities to the involved roles. This mapping of the different confidentiality levels can be managed e.g. following the Bell-LaPadula [1] approach. SSCM guarantees that **all security enforcing and security relevant objects ... shall be identified as such**[ITSEC]. SSCM rejects any unauthorized access to any CI. Subsequently each unauthorized access must be restricted or completely avoided and each of it has to be audited. The CI may be blocked until the SSCM administrator explicitly releases it. As this approach protects from malicious access it enables the SSCM to **ensure that the person responsible for acceptance of an object into configuration control was not one of its designers or developers**[ITSEC]. SSCM guarantees the separation of roles as a quality and security aspect in the development process.

The accountability is guaranteed by recording any action concerning the SSCM and especially any access to any CI. Any unauthorized access is marked separately and is analyzed by an automatic audit and report process (e.g. "red light" or warning at the console).

SSCM provides an efficient audit mechanism to get extreme transparency of the complete life cycle of any CI. SSCM must **control and audit changes between different versions of objects All modifications of these objects shall be audited with originator, date and time [ITSEC]**. Such an audit trail enables SSCM to manage the whole history of any CI. The responsibility of any change process is documented. At any time any malicious access or change to any CI shall be verified.

SSCM guarantees the accuracy of any CI and any released product version by providing a transparent and controlled change process. This approach is supported by the possibility of automatically rebuilding of any version of any CI or each released product. SSCM guarantees the accuracy and the integrity of the security kernel, regardless whether the kernel is concerned by the change process directly or indirectly. SSCM must **be able to identify all other objects ... affected by this change together with an indication of whether they are security enforcing or security relevant objects[ITSEC]**. Using this approach SSCM provides a special protection mode to guarantee the integrity of the security kernel of any software product.

The confidentiality, the integrity and the availability of all CIs and of the final released product versions will be ensured by SSCM following the discussed quality security aspects. Also additional topics as the limited reuse of highly confidential components may be managed by SSCM. Implementing the same security component only in a restricted number of software products will help to control the lack and its impact to all products using this component in the case of a successful penetration [12]. But this may work in contradiction to the economical aspects in SW engineering.

5. State of the Art

In current software engineering processes it may be distinguished between 5 maturity levels for software configuration management systems.

Level 1 is indicated by development processes without any configuration control. Such a process can not even guarantee the basic requirements of the ISO nor the ITSEC or the CC. Software products built by those processes should never be accepted in the scope of critical software in general and in the scope of high security software in special. A security evaluation of software products against the ITSEC or the CC would not be possible without any SCM.

Level 2 is indicated by a SCM without the application of a SCM tool. The SCM is managed completely by administrative activities. Such a SCM is characterized by high efforts to even guarantee the application versus the more basical quality and security requirements discussed in chapter 3. A non tool based SCM may be bypassed, ignored, or may be insufficient to prevent inconsistencies or unauthorized modifications. Hence a security evaluation of a product using a SCM level 2 may be restricted to a lower assurance level of the ITSEC or CC (e.g. E1 respectively EAL1 or EAL2). But evidence has to given for each software product for the implementation and application of this administrativeable SCM. This kind of evidence may require high efforts in respect to every quality assurance and to every security evaluation process.

Level 3 is indicated by a tool based SCM, e.g. in the CC it is subleveled by partial or complete automation. In general these tools guarantee a defined SCM in the development process fulfilling the basic quality and security requirements discussed in chapter 3. A tool based SCM

is a precondition for a successful security evaluation of higher assurance levels (beyond E4 of the ITSEC respectively EAL 5 in the CC). If once verified the results of the security evaluation concerning SCM may be reused with minimum effort for all products managed by this tool based SCM. Most of these tool based SCMs however don't fit the special security requirements discussed in chapter 4 and will not reach level 4 in our scale. Level 4 is indicated by such a tool based SSCM that completely fulfills the requirements explained in chapter 4. But even such level 4 SSCMs often dependent on the such security features of implemented environmental features as the underlying development operating system or the linked database.

Hence level 5 defines a SSCM of level 4 combined with full flexibility. Depending on the situation, the environment and the specific security related requirements the SSCM may be tailored including upgrading features in the sense of optimizing during the engineering process (e.g. including multilevel security (MLS) approaches). It has to be managed in a way independent of all the underlying or environmental SW-products as the operating system or the linked database. All the phases in the life cycle may be managed and controlled, the access control during the different phases may be adaptable from mandatory access control (MAC) to discrete access control (DAC). The reuse of highly confidential components has to be respected.

6. Future Aspects

In the development of high security software products as well as in the development of critical software in general, the evaluation and accreditation of the software development process including the infrastructure, the tools and the experience of the developers have become an established part of the software manufacturing process. Modern evaluation and accreditation approaches consider the impact of a sound development process on the security of an IT product/system. One basic part of this development process is SSCM. To minimize evaluation efforts future SSCMs should be independent of the specific operating systems and development environments. This approach enables a classification of the SSCM tools versus the requirements discussed in chapter 3 and 4 and will support the integration of the SSCM tools in modern developmental assurance approaches discussed in Europe and in USA. But as said in the introduction, the combination of technical aspects with the organisational and management issues will built a complete appropriate approach. This kind of a global approach is also valid for the SCM in IT-Security. So next steps are the interpretation and impacts from SCM to related issues in organisation and management.

The SSCM approach discussed in this paper currently finds high interest on the user and manufacturer side in the scope of high security software products, and leads to busy developmental activities of the tool manufacturers.

References

- [1] [BLP] Secure Computer Systems: Unified Exposition and Multics Interpretation, D.E. Bell and L.J. LaPadula, The Mitre Corporation, 1976
- [2] [CC] Common Criteria for Information Technology Security Evaluation - V1.0, CCEB, January 31, 1996
- [3] [CMM] Key Practices of the Capability Maturity Model, Mark C. Paulk, Charles V. Weber, Suzanne M. Gracia, Mary Beth Chrissis, Marilyn Bush, Software Engineering Institute, Pittsburgh, 1993
- [4] [CTCPEC] The Canadian Trusted Computer Product Evaluation Criteria V3.0e, Canadian System Security Centre - Communications Security Establishment - Government of Canada, 1993

- [5] [ESA] ESA Software Engineering Standards, Issue 2, ESA Board for Software Standardisation and Control, Paris, 1991
- [6] [ITSEC] Information Technology Security Evaluation Criteria, ECSC-EEC-EAEC, Brussels * Luxembourg, 1991
- [7] [ITSEM] Information Technology Security Evaluation Manual, ECSC-EEC-EAEC, Brussels * Luxembourg, 1994
- [8] [ISO] EN ISO 9000 Part 3: Guidelines for the application of ISO9001 to the development, supply and maintenance of software, NQZS, Beuth Verlag GmbH, Berlin, 1994
- [9] [MSWP] Managing the Software Process, Watts S. Humphrey, Addison-Wesley Publishing Company, 1989
- [10] [SCM] Software Configuration Management - Coordination for Team Productivity, Wayne A. Babich,
- [11] [SEPA] Software Engineering - A Practitioner's Approach, Roger S. Pressman, Mr Graw-Hill Book Company
- [12] [TSRR] Trusted Software, Repositories and Reuse, Mark O. Aldrich, GRC International, Proceedings of the 11. Annual Computer Security Application Conference, 1995 New Orleans, IEEE Computer Society Press, ISSN 1063, ISBN 0-8186-7159-9, page 208-216
- [13] [VMOD] Planung und Durchführung von IT-Vorhaben in der Bundesverwaltung - Vorgehensmodell, KBSt, Bundesanzeiger Verlagsgesellschaft mbH, Köln, 1992

THE DEPARTMENT OF DEFENSE INFORMATION TECHNOLOGY SECURITY CERTIFICATION AND ACCREDITATION PROCESS (DITSCAP)¹

Jack Eller

DISA, CISS (ISBEC)
701 South Courthouse Rd.
Arlington, VA 22204-4507

Mike Mastrorocco

Computer Security Consulting
107 Windsor Drive
Mineral Wells, WV 26150

Barry C. Stauffer

CORBETT Technologies, Inc.
228 N. Saint Asaph St.
Alexandria, VA 22314

Abstract

On August 19, 1992 the Office of Assistant Secretary of Defense directed the Defense Information Systems Agency (DISA) Center for Information Systems Security (CISS) to formulate a standard DoD process for security certification and accreditation. CISS formed a working group, consisting of Service and Agency representatives. The working group evaluated ten existing processes, but found none which could be adopted Department of Defense (DoD)-wide. As a result, the working group developed the DoD Information Technology Security Certification and Accreditation Process (DITSCAP) [1]. A standard process across DoD, DITSCAP applies to accreditation of both strategic and tactical systems, as well as stand-alone information systems or networks. DITSCAP capitalized on approved security techniques, software, and procedures to reduce the complexity and overall cost of the accreditation process. The DITSCAP integrates security directly into the system life cycle and is designed so that it can be applied uniformly across DoD. The DITSCAP defines a process which standardizes all activities leading to a successful accreditation, thereby minimizing the risks associated with nonstandard security implementations across shared Defense Information Infrastructure (DII) and end systems. The DITSCAP has been designed to support the requirements of Office of Management and Budget Circular A-130 [2].

In contrast to the prevailing system based accreditation processes, the DITSCAP is focused on the infrastructure and views systems and networks as components of the infrastructure. The view of the DITSCAP, therefore, differs from such documents as the National Computer Security Center (NCSC) Certification and Accreditation Process Handbook for Certifiers (NCSC-TG-031) [3]. CISS and the NCSC have agreed that for the near term, NCSC-TG-031 provides sound guidelines. DITSCAP provides the midterm and long term infrastructure-centric approach to the security certification and accreditation of systems and networks. These two processes have been harmonized to reflect the transition to the DITSCAP. Both terminology and structural parallels will facilitate a smooth transition between these two processes.

1. Introduction

The DITSCAP establishes a standardized process, set of activities, general task descriptions, and a management structure to verify, validate, implement and maintain the security posture of the DII. The DITSCAP is designed to be adaptable to any type of Information Technology (IT) and any computing environment and mission. It can be adapted to include existing system certifications and evaluated products. It can use new security technology or programs, and adjust to the appropriate standards. The process may be aligned with any program acquisition strategy. Its activities can be integrated into the system life cycle to ensure the system meets the accreditation requirements during development and integration and continues to maintain the accredited security posture after fielding. While DITSCAP maps to any system life cycle

¹ The DITSCAP was developed for CISS under Logicon, Inc. Contract DAAB07-91-D-B519

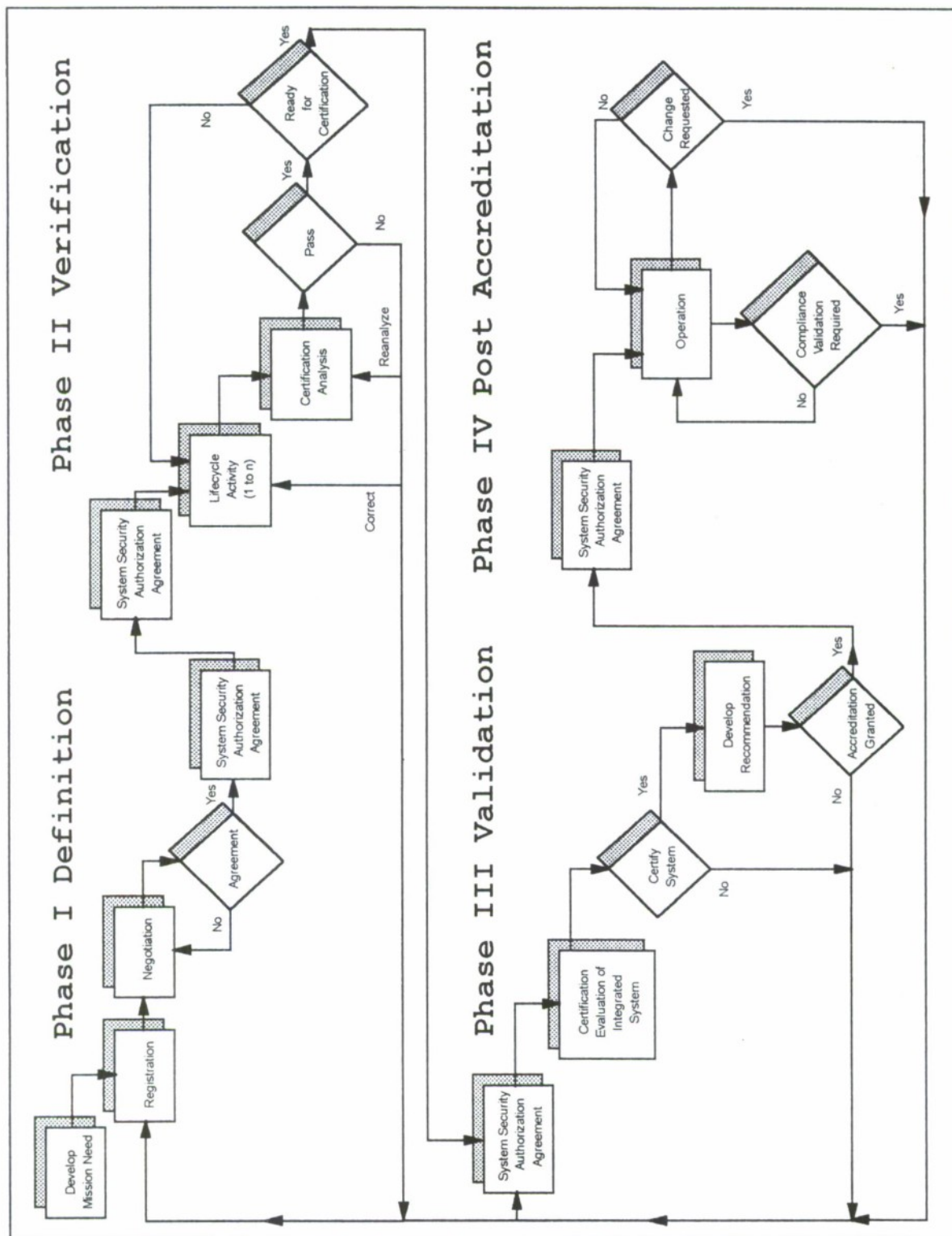


Figure 1. The DoD Information Technology Security Certification and Accreditation Process

process, its four phases are independent of the life cycle strategy. The DITSCAP's, four phases, Figure 1, are: Definition, Verification, Validation, and Post Accreditation. Phase I, **Definition**, focuses on understanding the mission, environment, and architecture to determine the security requirements and level of effort necessary to achieve accreditation. Phase II, **Verification**, verifies the evolving, or modified, system's compliance with the agreed upon security requirements. Phase III, **Validation**, validates the fully integrated system's compliance with the security requirements. Phase III concludes with full approval to operate the system, e.g., security accreditation. Phases I, II, and III are the DITSCAP process engine. The DITSCAP methodology permits the forward or backward movement between phases to keep pace with the system development or to resolve problems. Therefore the phases are repeated as often as necessary to produce an accredited system. The objective of Phase IV, **Post Accreditation**, is to ensure system management, operation, and maintenance to preserve an acceptable level of residual risk. Phase IV includes those activities necessary for the continuing operation of the accredited system.

Each phase is performed for every system and every process activity within each phase is performed. However, the procedures within each process activity may be tailored and scaled to the system and its associated acceptable level of residual risk. The procedures are a set of established tasks which can be tailored to fit the mission, environment, system architecture, and programmatic considerations. These procedures consist of planning, certification, development, maintenance, operation, change management, and compliance validation actions. In this manner, the process maintains flexibility to deal with different acquisition strategies, situations, and operational scenarios.

2. Phase I Definition

Phase I activities focus on definition of the certification and accreditation task. This is the planning phase which documents all results in the System Security Authorization Agreement (SSAA). Phase I is similar to other certification and accreditation processes in that the planning is begun, appropriate security officials are identified, responsibilities are assigned, data is collected and a security plan is initiated. Unlike other processes, Phase I ends with a formal agreement of the definition of the architecture and boundaries of the system to be certified, security requirements, certification approach, work plan, and level of effort. Phase I is not completed until this agreement is reached. Phase I is revisited throughout the process, whenever necessary, to update this agreement.

The key to the DITSCAP is the agreement which is reached in Phase I between the IT system Program Manager, the Designated Approving Authority (DAA), and the User Representative. These three managers resolve critical schedule, budget, security, availability, functionality, and performance issues and document that agreement in the SSAA.

Phase I contains three process activities; Mission Need, Registration, and Negotiation. The input to phase I includes all available system documentation, security requirements, system requirements, and the Concept Of Operations. The output from Phase I is the SSAA. The three process activities provide the pathway to understanding the system; documenting the security requirements; developing a security architecture; and determining the scope, level of effort, documentation required, and schedule for the planning and certification actions. Phase I begins with analyzing or developing the mission need. The mission need is either a document or compilation of information which state the systems requirements and intended capabilities. It includes the definition of the system mission, functions and interfaces; organization(s) to operate the system; the intended operational environment; information types and classifications;

expected system life cycle; system user characteristics; and intended interfaces with other systems of networks. As implied by the process activity name (mission need), the DITSCAP starts as soon as the concept for a system is developed. If the system of interest is an existing system, the process starts when a security relevant modification is being planned, or upon the periodic reaccreditation.

Registration starts the dialogue between the Program Manager, the DAA, and the Users of the system. The Program Manager and the DAA appointed Certification Authority work together to perform the certification actions. During Registration, information is collected and evaluated, applicable security requirements are determined, risk management and vulnerability assessment activities begin, and the level of effort required for certification and accreditation is determined and planned. Registration begins with a review of the mission need and concludes with preparation of an initial draft of the SSAA. These activities involve the collection of necessary information to determine security requirements and the level of effort to accomplish the certification and accreditation commensurate with the level of assurance required for confidentiality, integrity, availability, and accountability. The results of the Registration activities are documented in the SSAA. The draft SSAA is then submitted to the Program Manager, DAA, and User Representative for their review. The tasks required during Registration activities include:

- Prepare a high level system description including system boundaries and interfaces.
- Determine the system program acquisition strategy and life cycle of the system.
- Assess the impact of the system life cycle phase on the certification effort.
- Determine the classification and types of information to be processed.
- Determine the clearances and access requirements of the processes and users.
- Identify the system class and develop the system security requirements.
- Identify the organizations that will support the DITSCAP.
- Tailor the DITSCAP activities.
- Determine the scope, level of effort and schedule for the DITSCAP activities.

Negotiation is the activity where all the participants involved in the information system's development, acquisition, operation, security certification, and accreditation agree upon the implementation strategy to be used to satisfy the security requirements identified during system registration. The key parties who must reach agreement during the negotiations are the Program Manager, the DAA, and the User Representative. Negotiation is NOT a bargaining session to determine which requirements to implement and which to delete. The purpose of negotiation is to ensure that all participants understand their roles and responsibilities and that the SSAA properly and clearly defines the requirements, the approach, and the level of activity. Negotiation concludes with the approval of the SSAA by the Program Manager, DAA, and User Representative.

The SSAA documents the conditions of certification and accreditation for an IT system. The SSAA is used throughout the entire DITSCAP to guide activities, document decisions, specify security requirements, document certification tailoring and level-of-effort, identify potential solutions, and maintain operational systems security. The SSAA is a "living", master document intended to reduce the need for repetitive documents by consolidation of all security related information into one document. The SSAA is the baseline reference for future decisions. As such it is particularly helpful during personnel changes and program budget modifications.

3. Phase II Verification

The activities of Phase II, verify the system's compliance with the requirements agreed on in the SSAA. Phase II activities include continuing refinement of the SSAA, system development or modification, and certification analysis. Phase II starts with a review and, if necessary, refinement of the SSAA. At each stage of the development or modification, the SSAA is refined by adding details to reflect the current state of the system. As the development or modification progresses and specific information relating to the certification effort becomes available, the SSAA is updated to include more specific details. As details about the hardware and software architecture become available, this information is added to the SSAA to support the agreed upon level of certification actions. Since the Program Manager, DAA, and User Representative concur with all changes of the SSAA, they are continually appraised on the security requirements, DITSCAP activities, and level-of-effort. As a result there are no surprises at certification and accreditation time.

The life cycle activities in Phase II are those activities required to develop and integrate the system components. Each life cycle activity has a corresponding Phase II certification analysis activity. These certification activities verify by analysis, investigation, and comparison that the IT design implements the SSAA requirements and the security critical components function properly. Certification analysis complements the functional testing which occurs during Phase III. While every system may be considered certifiable, the DITSCAP goal is to produce systems that satisfy operational requirements with an acceptable level of risk. The DITSCAP analysis actions, therefore, are performed in step with the system development to ensure the development, modification, and integration efforts will result in a certifiable and accreditable information system, before Phase III begins. In this manner, DITSCAP becomes a success oriented process.

Six² certification actions or tasks are performed during Phase II. These include:

- System Architecture analysis verifies that the system architecture complies with the architecture description agreed upon in the SSAA.
- Software Design analysis evaluates how well the software implements the security requirements of the SSAA and the security architecture of the system.
- Network Connection Rule Compliance analysis evaluates connections to other systems and networks to ensure the system design will enforce security policies.
- Product integrity analysis evaluates the integration of non-developmental software, hardware, and firmware to ensure their integration complies with the system security architecture, and the integrity of each product is maintained.
- Life Cycle Management analysis verifies that change control and configuration management practices are, or will be, in place and are sufficient.
- Vulnerability Assessment evaluates security vulnerabilities and recommends appropriate countermeasures.

At the completion of the Phase II Certification Analysis, the system will have a documented security specification, a comprehensive test plan, and assurance that all network and other

²The 21 INFOSEC analysis tasks described in the "Certification and Accreditation Process Handbook for Certifiers", NCSC-TG-031, have been restructured into 14 tasks in the DITSCAP.

interconnections requirements have been implemented. A vulnerability assessment will have been conducted and will have concluded that the infrastructure needs of the system, e.g., configuration management, will be accommodated throughout the system life cycle.

At the conclusion of each life cycle development milestone, the certification analysis results are reviewed for SSAA compliance. Should the results indicate significant deviation from the SSAA, the DITSCAP reverts to Phase I to resolve the problems. If the results are acceptable, the DITSCAP proceeds to the next development activity or to government acceptance and security testing, i.e., DITSCAP Phase III. Upon completion of certification analysis, the system proceeds to Phase III, which contains the formal system certification test and security accreditation actions.

4. Phase III Validation

Phase III activities, Figure 1, validate that preceding work has produced a system that operates in a specified computing environment with an acceptable level of residual risk. Phase III begins with a review of the SSAA to ensure that its requirements and the agreements are current. The SSAA review is followed by an evaluation of the IT system, certification, and accreditation. Phase III activities occur after the system is integrated and culminate in system accreditation.

Certification Evaluation includes eight actions to certify that the fully integrated system is ready for operational deployment. These actions include:

- System Security Testing and Evaluation to assess the technical and nontechnical implementation of the security features and their proper performance.
- Penetration Testing, for appropriate system classes, to assess the system's ability to withstand attempts to circumvent system security features.
- TEMPEST and Red/Black Verification to validate that the equipment and site meet the applicable TEMPEST security requirements.
- Validation of COMSEC Compliance to ensure that COMSEC approval has been granted and approved COMSEC key management procedures are used.
- System Management Analysis to examine the management infrastructure to determine if it will maintain the mission, environment, and architecture described in the SSAA.
- Site Accreditation Surveys to validate that the operational procedures for the IT, environmental concerns, and physical security pose no unacceptable risks.
- Contingency Plan examination to verify that the contingency and continuity of service plans are consistent with the SSAA.
- Risk Management Review to assess the operation of the system to determine if the risk is being maintained at an acceptable level.

At the conclusion of Phase III, the certifier prepares a recommendation to the DAA. The recommendation, supporting documentation, and the SSAA form the accreditation package. The supporting documentation should include security findings, deficiencies, risks of operation and all information necessary to support the recommended decision. After the accreditation

decision is made, the system now progresses to Phase IV.

5. Phase IV Post Accreditation

Phase IV contains activities necessary to operate and manage the system so that it will maintain an acceptable level of residual risk. Post-accreditation activities include; ongoing maintenance of the SSAA, system operations, change management, and compliance validation. Phase IV begins after the system has been integrated into the operational computing environment and accredited. Phase IV continues until the information system is removed from service. As in the preceding phases, the SSAA must be kept current.

The second Phase IV activity, Operation, concerns the secure operation of the system and the associated computing environment. System maintenance activities ensure the system continues to operate within the stated parameters of the accreditation. These activities identify changes in hardware, software, and system design. The system security officer determines the extent to which a change affects the security posture of either the information system or the computing environment. Changes that significantly affect the system security posture must be forwarded to the DAA, User Representative, and Program Manager. In this manner, the system continues to operate under Phase IV while the proposed changes are considered under Phase I of the DITSCAP. The three managers then decide what certification and accreditation actions are required in response to the proposed change.

Compliance Validation is a periodic review of the operational system and its computing environment to ensure the continued compliance with the security requirements, current threat assessment, and concept of operations as stated and agreed upon in the SSAA.

6. Process Management Roles and Responsibilities

The management approach for DITSCAP focuses on systems level management to execute DITSCAP. The management concept integrates existing roles into the certification and accreditation process to provide visibility into the process to all managers responsible for system development, operation, maintenance, security, and to system users.

There are three key roles in the DITSCAP, the system Program Manager, the DAA, and the User Representative. These three managers cooperate to provide the most capable system with an acceptable (tolerable) level of risk. They develop and approve the security requirements, manage the certification and accreditation process, and review the results. They must reach agreement during Phase I "Negotiation" and approve the SSAA. During Phases II, III, and IV, if the system is changed, or any of the agreements delineated in the SSAA are modified, the three key parties return to Phase I Negotiation and revise the SSAA as necessary. The DITSCAP allows these three managers to tailor and scope the certification and accreditation efforts to the particular mission, environment, system architecture, threats, funding, schedule, and criticality of the system.

7. Summary

The DITSCAP provides the common framework to certify and accredit all DoD IT systems within the network infrastructures they employ and to maintain the security of these systems throughout their life cycle. While the activities in the four DITSCAP phases are mandatory, the

implementation details may be tailored to meet the need of the particular system.

Potential savings are anticipated for all organizations involved in DoD certification and accreditation. Reuse of security architectures, designs, or certification evidence is facilitated by categorizing systems into a set of system classes. The SSAA consolidates and reduces documentation requirements eliminating the repetition of information in multiple documents. The use of the standard process should eliminate duplicate efforts and promote common acceptance of data generated by different agencies and DAAs. The use of system classes facilitates the sharing of results.

References

- [1] Department of Defense (DoD) Information Technology Security Certification and Accreditation Process (DITSCAP), July 1996.
- [2] Office of Management and Budget (OMB) Appendix III to OMB Circular No. A-130 - Security of Federal Automated Information Resources, February 1996.
- [3] National Computer Security Center (NCSC) Certification and Accreditation Process Handbook for Certifiers (NCSC-TG-031), July 1996.

TRUSTED PROCESS CLASSES

William L. Steffan
Tracor Applied Sciences, Incorporated
503A Coliseum Boulevard
Montgomery, Alabama 36109
Voice: 334-271-6804 FAX: 334-244-0058
wsteffan@b856s1.ssc.af.mil

Jack D. Clow
SenCom Corporation
6004C East Shirley Lane
Montgomery, Alabama 36117
Voice: 334-277-1972 FAX: 334-277-1932
jclow@b856s1.ssc.af.mil

ABSTRACT

Vendors who develop Trusted Computing Base (TCB) equipped secure operating systems face difficult choices as they design and implement the requisite protection features appropriate to the evaluation class being targeted (e.g., *Labeled Security Protection*, Class B1). On the one hand, vendors seek to meet each and every evaluation class requirement unconditionally, being careful to limit every possible opportunity for latent vulnerabilities to occur. However, on the other hand, vendors must not implement their secure product with so many constraints that it loses its competitive advantage and utility as an operating system having general applicability throughout the marketplace. Balancing these conflicting goals often results in the vendor's implementing a *more restrictive* rule set than permitted by theoretical considerations.

Unfortunately, unlike the published criteria for TCB classes themselves¹, developers who implement trusted processes have had to depend on *ad hoc* experimentally derived guidelines and rules to meet both mission and security requirements simultaneously.

This paper presents a new methodology, derived from the theory of a TCB-equipped operating system and practical experience, to explicitly determine to which of several classes a specific trusted process* belongs; as well citing applicable programming confinement rules to ensure additional risks, if any, will be acceptable.

* A trusted process is a program, module, or algorithm which has extraordinary privilege(s), which if not otherwise strictly controlled and limited, could subvert the security policy in unpredictable ways -- in the extreme, subvert the protection domain provided by the TCB for itself.

Keywords: Trusted Computing Base, trusted process, mandatory access controls, discretionary access controls, auditing, certification, accreditation.

TRUSTED PROCESS CLASSES

1. **Background.** Vendors who develop Trusted Computing Base (TCB) equipped secure operating systems face difficult choices as they design and implement the requisite protection features appropriate to the evaluation class being targeted (e.g., *Labeled Security Protection*, Class B1). On the one hand, vendors seek to meet each and every evaluation class requirement unconditionally, being careful to limit *every possible* opportunity for latent vulnerabilities to occur. However, on the other hand, vendors must not implement their secure product with so many constraints that it loses its competitive advantage and utility as an operating system having general applicability throughout the marketplace. Balancing these conflicting goals often results in the vendor's implementing a *more restrictive* rule set than permitted by theoretical considerations. For example, while formal TCB theory would permit a *high-clearance* program (i.e., a *subject*) to read a *low-sensitivity* data item (i.e., an *object*), some vendors enforce a *clearance SHALL ALWAYS EQUAL sensitivity* rule rather than the more general *clearance SHALL BE GREATER THAN OR EQUAL TO sensitivity* rule permitted by theory.

Thus, for these and other reasons, TCB-equipped operating systems usually fall short of providing security protection features needed to support *every possible* customer application. In turn, then, software developers who use these TCB-equipped operating systems to meet customer mission-oriented applications face some awkward possibilities:

- Failing to meet the customer's mission requirements since the vendor's TCB-equipped operating system and its protection features prohibits achieving some essential mission functionality.
- Failing to meet the customer's security requirements since the vendor's TCB-equipped operating system does not provide the necessary features to protect some essential mission functionality and/or sensitive information.
- Failing to meet both the customer's mission and security requirements since the vendor's TCB-equipped operating system implements a more restrictive enforcement policy than that permitted by theory.

Classically, the solution has been to incorporate developer-written *trusted processes* so that

- Customer mission requirements will be met unconditionally,
- While accepting some additional risk by permitting local controlled deviations to otherwise globally applicable security policy enforcement rules.

Unfortunately, unlike the published criteria for TCB classes themselves¹, developers who implement trusted processes have had to depend on *ad hoc* experimentally derived guidelines and rules to meet both mission and security requirements simultaneously.

2. **Purpose.** This paper presents a new methodology, derived from the theory of a TCB-equipped operating system and practical experience, to explicitly determine which of several

classes a specific trusted process belongs to. It also cites applicable programming confinement rules to ensure that any additional risks will be acceptable.

3. Definitions. Definitions for key concepts and terms used herein follow.

a. Discretionary Access Controls. *Discretionary Access Controls (DAC)* are rules enforced by the *reference monitor* which provide for need-to-know violation prevention as prescribed by the security policy for the system (i.e., an entity's functional role is a sufficient basis for permitting access to system-protected resources).

b. Integrity Principle. A fundamental underlying assumption of a TCB-equipped operating system that states a *high-integrity* entity (e.g., the *reference monitor*) shall NEVER "trust" assertions made by a *lower-integrity* entity (e.g., an untrusted applications program), but a *low-integrity* entity shall ALWAYS "trust" assertions made by a *higher-integrity* entity.

c. Least Privilege Principle. A fundamental mandate for a TCB-equipped operating system that states every entity shall be granted ONLY the MINIMUM privilege(s) essential to perform its assigned function(s) and NO MORE.

d. Mandatory Access Controls. *Mandatory Access Controls (MAC)* are rules enforced by the *reference monitor* which provide for compromise prevention as prescribed by the security policy being enforced for the system (e.g., an entity's clearance is a sufficient basis for permitting access to system-protected resources).

e. Reference Monitor. A *reference monitor* is that portion of a TCB-equipped operating system having exclusive responsibility for enforcing *Discretionary Access Controls* and *Mandatory Access Controls* according to mathematically precise rules.

f. Tranquillity Principle. A fundamental underlying assumption of the Bell-LaPadula formal security model for a *reference monitor* which states that, once identified to the *reference monitor*, the sensitivity level contained in a subject's *sensitivity label* (or an object's *sensitivity label*) shall remain invariant unless explicitly changed under the express control of the *reference monitor*.

g. Trusted Process.

(1) A *trusted process* is a program, module, or algorithm written expressly by a developer that has these characteristics:

- May require over-riding security policy enforcement mechanisms or their underlying assumptions.
- Does not subvert security policy mandated rules except in explicitly controlled ways in a constrained local context.
- NEVER enforces globally applicable security policy mandated rules.

(2) A *trusted process* is a program, module, or algorithm which has extraordinary privilege(s), which if not otherwise strictly controlled and limited, could subvert the security

policy in unpredictable ways -- in the extreme, subvert the protection domain provided by the TCB for itself.

4. Methodology. This section examines some relevant factors to determine the *Trusted Process Class* for any given trusted process.

a. Trusted Process Observations. In practice, as suggested by themes given in *TCB Subjects - Privileges and Responsibilities* in [2], trusted processes may be granted privileges which over-ride enforcement rules for DAC, MAC, Tranquillity Principle, or any combination thereof. For example:

- To produce an unclassified report about the existence of, but not the value of, some classified fact, a *low-clearance* trusted process subject needs to read, but not reveal, the information content in a *high-sensitivity* object. To do this, the trusted process subject must temporarily over-ride MAC enforcement rules.
- To perform a regrade on some imported file, a *high-clearance* trusted process subject must change the sensitivity label *content* (i.e., the sensitivity level), but not the information content in the object itself, for a *low-sensitivity* object. To do this, the trusted process subject must temporarily over-ride the Tranquillity Principle.
- To perform an emergency recovery action, a trusted process subject must be granted temporary *execution privilege* over-riding DAC enforcement rules.
- To provide standard agency-specified security markings on human readable media, a trusted process subject must intercept and faithfully translate internally coded binary representations for the sensitivity label content. To do this, the trusted process subject must uphold the *Integrity Principle*, while not subverting DAC, MAC, Tranquillity Principle, or other security policy-prescribed rules.

b. Trusted Process Classes. The cited examples suggest a methodology to determine the specific trusted process class for any given trusted process. As Figure 1 illustrates, these trusted process classes can be enumerated according to whether the trusted process must over-ride rules for DAC, MAC, Tranquillity Principle, or combinations thereof.

Trusted Process Class	Over-Ride Privilege Granted			Permitted Action(s)
	Tranquillity Principle	Mandatory Access Controls	Discretionary Access Controls	
0	---	---	---	Read, Write, or both Read & Write
1	---	---	Yes	
2	---	Yes	---	
3	---	Yes	Yes	
4	Yes	---	---	
5	Yes	---	Yes	
6	Yes	Yes	---	
7	Yes	Yes	Yes	
	<i>Label</i>	<i>Content</i>	<i>Privileges</i>	

Figure 1, Trusted Process Classes

The last row in Figure 1 lists some supplementary guidelines to help determine the appropriate trusted process class. For example, if a trusted process supports a regrade capability by changing ONLY the sensitivity label *content* (or sensitivity level), it needs to over-ride the Tranquillity Principle ONLY and is, therefore, a Class 4 Trusted Process. In a similar fashion, a trusted process supporting a program producing an unclassified report about the existence of some classified fact, but not the value of the fact, needs to over-ride MAC enforcement by examining the content of some object -- a Class 2 or Class 3 Trusted Process.

The right-most column in Figure 1 gives the potential operations a trusted process class might perform. Thus, a Class 4 Trusted Process may have to both read and write to the sensitivity label for an object (or subject) as it operates with Tranquillity Principle over-ride privileges. Note the Class 0 Trusted Processes are special cases which require no over-ride privilege, but are, accordingly, restricted to *read only* operations -- as would be needed to translate the binary sensitivity label content used in the TCB-equipped operating system itself to the binary sensitivity label content used in the TCB-equipped database management system.

Figure 1 also illustrates that, as the Trusted Process Class *number* increases, so do the risks associated with using trusted processes in that class. This is especially true for Trusted Process Classes 6 and 7 where, unless there are compelling mission satisfaction reasons involved, the risks may be unacceptably high.

Finally, using the *Integrity Principle*, it is easy to show that as the Trusted Process Class *number* increases, the inherent *trustworthiness* of each class decreases since the potential for "abuse" increases. Thus, trusted processes which do not over-ride the *Tranquillity Principle* are more trustworthy than those which do.

5. **Trusted Process Implementation.** In the text above, the phrases "... *the potential operations a trusted process class might have to perform.*" and "... *may have to both read and write to the sensitivity label ...*" are, regrettably, ambiguous. Resolving such ambiguity demands that the *Least Privilege Principle* be explicitly invoked for *each* and *every* trusted process. Moreover, there are MANDATORY programming confinement rules that must be carefully followed.

a. Trusted Process Programming Confinement Rules (PCR). This section cites the programming confinement rules that a trusted process developer MUST obey.

(1) Local Domain Context Storage [PCR-1]. *Each* and *every* trusted process shall use local domain context storage ONLY for variables used in the trusted process.

(2) Local Domain Context Storage Purge [PCR-2]. For *each* and *every* trusted process, the last step prior to exiting from the trusted process shall purge (i.e., set to all binary zero) *all* variables used in the trusted process.

(3) Trusted Process Audit [PCR-3]. *Each* and *every* trusted process invocation shall be auditable³ by the TCB-equipped operating system per the schedule in Table PCR-1.

Table PCR-1, Trusted Process Audit Requirements

Trusted Process Class	Audit Required	Remarks
0	Optional	May be selectively audited
1	Optional	May be selectively audited
2	Optional for Read ALWAYS for Write	---
3	Optional for Read ALWAYS for Write	---
4	ALWAYS	No Exceptions Permitted
5	ALWAYS	No Exceptions Permitted
6	ALWAYS	No Exceptions Permitted
7	ALWAYS	No Exceptions Permitted

(4) Assignment Statement Restrictions [PCR-4]. For MAC over-ride privileged trusted processes, the *value of the object read* shall NEVER appear ALONE on the right-hand side of an expression in an assignment statement.

(5) Function or Subroutine Return Parameter Restrictions [PCR-5]. For MAC over-ride privileged trusted processes, the *value of the object read* shall NEVER be used as the return value for the trusted process function or subroutine.

(6) Least Privilege Principle Restrictions [PCR-6]. *Each and every* trusted process shall ONLY use the MINIMUM privilege(s) from the over-ridden set to perform its function.

(7) Computational Expression Restrictions [PCR-7]. For MAC over-ride privileged trusted processes, the *value of the object read* may be used in a computational expression provided that the *value of the object read* shall NEVER be revealed to any entity outside the local domain of the trusted process itself.

(8) Logical Expression Restrictions [PCR-8]. For MAC over-ride privileged trusted processes, the *value of the object read* may be used in a logical expression provided that the *value of the object read* shall NEVER be revealed to any entity outside the local domain of the trusted process itself.

(9) Single Functionality Restrictions [PCR-9]. *Each and every* trusted process shall perform a SINGLE well-defined function.

(10) Single Entry Restrictions [PCR-10]. *Each and every* trusted process shall have a SINGLE well-defined entry point for execution to begin.

(11) Single Exit Restrictions [PCR-11]. *Each and every* trusted process shall have a SINGLE well-defined exit point for execution to conclude.

(12) Trusted Process Author [PCR-12]. *Each and every* trusted process shall have an appropriately cleared assigned author whose work shall be INDEPENDENTLY verified by an appropriately cleared analyst (e.g., for a SECRET system, a SECRET cleared person).

(13) Configuration Management Handling Restrictions [PCR-13]. *Each and every* trusted process shall be assigned to a "Trusted Process Library" with access restricted to specifically named persons ONLY.

(14) Trusted Process Qualification Testing [PCR-14]. *Each and every* trusted process shall be tested comprehensively and the test results explicitly addressed in the *Security Test and Evaluation (ST&E) Report*.

b. Trusted Process Programming Confinement Rule Assignment Schedule. Table PCR-2 gives the schedule for applying the MANDATORY programming confinement rules cited in this section.

Table PCR-2, Trusted Process Programming Confinement Rule Assignments

Trusted Process Class	Rule Assignment(s)			Universally Applicable PCRs
	Tranquillity Principle	Mandatory Access Controls	Discretionary Access Controls	
0				1, 2, 3, 6, 9, 10, 11, 12, 13, 14
1				
2		4, 5, 7, 8		
3		4, 5, 7, 8		
4				
5				
6		4, 5, 7, 8		
7		4, 5, 7, 8		
	<i>Label</i>	<i>Content</i>	<i>Privileges</i>	

6. **Summary.** The need for trusted processes appears to circumvent the basic philosophy for using a TCB-equipped operating system in the first place -- why expend valuable resources for a secure *Trusted System* and then permit "deviations" to occur with trusted processes?

The facts are that developers will use trusted processes to meet mission imperatives in a TCB-enriched environment. Having done so, there is an inherent responsibility to show these trusted processes exhibit "trustworthiness" in a justifiable way.

By suggesting ways to explicitly deal with "trustworthiness," the methodology given in this paper fosters compliance with both *theory* and *practicality*. Moreover, its techniques can help in understanding the distinctions among the several types of trusted processes likely to be encountered in the "real world" -- some are relatively benign, others can entail serious, if not potentially catastrophic threats. The paper also suggests practical programming confinement rules which limit permitted actions a trusted process class can or should take thus providing a basis for assessing associated risks.

In a broader sense, "security bit-meisters" and system developers need to assure senior management that their investment in a secure system reflects a sound business decision. We believe the concepts developed in this paper can provide a common frame of reference for these necessary assurances.

References

- [1] *DoD Trusted Computer System Evaluation Criteria*, DOD 5200.28-STD, December 1985.
- [2] *A Guide to Understanding Security Modeling in Trusted Systems*, NCSC-TG-010, Version 1, October 1992.
- [3] *A Guide to Understanding Audit in Trusted Systems*, NCSC-TG-001, Version 2, June 1988.

Acknowledgment

Comments provided by the referees pointed out where we needed to hone and refine several important concepts; we thank them for their valuable insight and helpful suggestions.

Trusted Process Classes

19th National Information Systems Security Conference

**William L. Steffan
Tracor Applied Sciences, Incorporated
334-271-6804**

14/06/98 11:57

Slide 1 of 9

Trusted Process Classes

Overview

- **BACKGROUND**
- **METHODOLOGY MOTIVATION**
- **DEFINITIONS**
- **TRUSTED PROCESS CLASS DISCRIMINATORS**
- **IMPLEMENTATION CONFINEMENT RULES**
- **SUMMARY**

14/06/98 11:57

Slide 2 of 9

Trusted Process Classes

BACKGROUND

- **VENDOR DILEMMA**
 - Marketplace Pressures
 - NCSC Evaluation Hurdles
- **DEVELOPER DILEMMA**
 - Achieve Security Policy Compliance
 - Meet Operational Mission Requirements
- **ASSURANCE DILEMMA**
 - Trusted Process “Trustworthiness”
 - Common Framework for Understanding

14/06/98 11:57

Slide 3 of 9

Trusted Process Classes

Methodology Motivation

- **ADDRESS DILEMMA PROBLEMS**
- **CODIFY “LESSONS LEARNED”**
- **PROMOTE COMMON UNDERSTANDING**
- **BRIDGE BETWEEN THEORY AND PRACTICE**
- **MAKE RULES EXPLICIT**

14/06/98 11:57

Slide 4 of 9

Trusted Process Classes

DEFINITIONS

- **TRUSTED PROCESS. A PROGRAM OR ALGORITHM WITH THESE CHARACTERISTICS:**
 - **May over-ride security policy enforcing mechanisms**
 - **Does not subvert security policy rules except in explicitly controlled, locally constrained ways**
 - **NEVER enforces globally applicable security policy rules**

14/06/96 11:57

Slide 5 of 9

Trusted Process Classes

Definitions (continue)

- **TRANQUILLITY PRINCIPLE. DIGITAL SECURITY LABEL REMAINS INVARIANT**
- **DISCRETIONARY ACCESS CONTROLS (DAC). "NEED-TO-KNOW" RULES**
- **MANDATORY ACCESS CONTROLS (MAC). "CLEARANCE" VERSUS "CLASSIFICATION" COMPROMISE PREVENTION RULES**

14/06/96 11:57

Slide 6 of 9

Trusted Process Classes

TRUSTED PROCESS CLASS DISCRIMINATORS

TP Class	OVERRIDE PRIVILEGE GRANTED			Action Permitted
	Tranquillity	MAC	DAC	
0	---	---	---	Read Only
1	---	---	YES	R or W, R & W
2	---	YES	---	R or W, R & W
3	---	YES	YES	R or W, R & W
4	YES	---	---	R or W, R & W
5	YES	---	YES	R or W, R & W
6	YES	YES	---	R or W, R & W
7	YES	YES	YES	R or W, R & W

14/05/98 11:57

Slide 7 of 9

Trusted Process Classes

IMPLEMENTATION CONFINEMENT RULES

- PCR-1 Local Domain Context Storage
- PCR-2 Local Domain Context Storage Purge
- PCR-3 Trusted Process Audit
- PCR-4 Assignment Statement Restrictions
- PCR-5 Function or Subroutine Return Parameters
- PCR-6 Least Privilege Principle Restrictions
- PCR-7 Computational Expression Restrictions
- PCR-8 Logical Expression Restrictions
- PCR-9 Single Functionality Restrictions
- PCR-10 Single Entry Restrictions
- PCR-11 Single Entry Restrictions
- PCR-12 Trusted Process Author
- PCR-13 Configuration Management Restrictions
- PCR-14 Trusted Process Qualification Testing

14/05/98 11:57

Slide 8 of 9

Trusted Process Classes

Summary

- **DEFINE TRUSTED PROCESS CLASSES**
- **TRUSTWORTHINESS MADE EXPLICIT**
- **CONFINEMENT RULES FOR IMPLEMENTORS**
- **ASSURANCES**
 - Bridge between THEORY and PRACTICE
 - Meet SECURITY and MISSION REQUIREMENTS
 - Common understanding between SECURITY BIT-MEISTERS and SYSTEM DEVELOPERS
- **BOTTOM LINE: TRUST ... BUT VERIFY**

Design Analysis in Evaluations Against the TCSEC C2 Criteria

Frank Belvin
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
fb@mitre.org

Deborah Bodeau
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
dbodeau@mitre.org

Shaan Razvi
The MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420
srazvi@mitre.org

Abstract

This paper provides an overview of design analysis, which is a key component of product evaluations under the Trusted Products Evaluation Program (TPEP) and the emerging Trust Technology Assessment Program (TTAP). It describes activities performed and issues considered for evaluation against the TCSEC C2 criteria. This description is intended to provide a common understanding among the vendor and user communities of the design analysis process at the C2 level and to serve as an input to the emerging Common Evaluation Methodology for the Common Criteria.

1: Introduction

This paper provides an overview of design analysis activities for evaluation against the TCSEC C2 criteria [1] under the Trusted Products Evaluation Program (TPEP). This document is intended to capture current practice and lessons learned from past experience in evaluating products against the C2 criteria of the Trusted Computer System Evaluation Criteria (TCSEC). It should be viewed as a product of the evaluation community, with the understanding that the opinions and experience of individual evaluators and evaluation teams could differ from some of the material contained herein. The authors have incorporated material from many sources, including the TCSEC, its interpretations and guidelines, forum entries, the Form and Content documents, and management and process documents. We gratefully acknowledge comments from many members of the evaluation community.

This description is intended to improve understanding in the vendor and customer communities about what design analysis means in the context of C2. It is intended to serve as input to the emerging Trust Technology Assessment Program (TTAP).[2]. Finally, this description is intended to serve as input to the emerging Common Evaluation Methodology (CEM) under the Common Criteria (CC) [3].

The description of the design analysis process presented below is taken from the Derived Verification Requirements for Controlled Access Protection, which is being used in the first experimental evaluation by a commercial enterprise [2]. The description is based on [4-13] and written evaluator inputs. (While [4] describes evaluation activities for B2, much of its discussion is applicable to C2.) The focus is on the goals, key concepts, and questions that drive the design analysis process.

2: Overview

Design analysis is a major component of trusted product evaluation. The goal of design analysis is for the evaluation team to understand the intended use, philosophy of protection, Trusted Computing Base (TCB) architecture, and TCB interfaces of the target of evaluation (TOE). The team's understanding must be of sufficient depth so that the team can explain these topics to an audience which is unfamiliar with the TOE but has a deep understanding of the C2 criteria and how products have met or failed to meet those criteria. Under TPEP, this audience is the Technical Review Board (TRB), and they must understand how, if the TOE's implementation is consistent with the Philosophy of Protection (POP), Architecture Summary Document (ASD), and Interface Summary Document (ISD) on which the design analysis was based, the TOE will meet the requirements stated in the TCSEC C2 criteria, as amplified and clarified by official interpretations. The team's understanding must be of sufficient depth so that the team can justify to the TRB the effort of proceeding to test.

The team will also use the understanding it gains from design analysis to analyze the vendor's test coverage and to develop independent tests. Test coverage analysis can be performed concurrently with design analysis. Evaluation teams often uncover discrepancies between a vendor's design documentation and test documentation.

The current process is based on the assumption that the vendor will follow the Form and Content (F&C) Guidelines [12, 13]. In the description below, it is assumed that the vendor's design documentation consists of a Philosophy of Protection document, an Architecture Summary Document, and a TCB Interface Summary Document, as described in [12], and on all detailed design

documentation to which these documents point. The vendor provides the team with evidence which, in the vendor's judgment, proves that the TOE meets the C2 criteria. Deviations from the F&C Guidelines may be acceptable. The evaluation team may allow deviations in form, for example, by accepting architectural and detailed design documents from the vendor's documentation suite together with a "road map" that identifies, for each topic identified in [12], where the corresponding material can be found in the vendor's documentation. However, if the evaluation team allows deviations in form, the vendor must be prepared for the evaluation to take longer and for the evaluation team to request more supplementary material than if the design documentation had been prepared in accordance with Vendor Design Documentation [VDD]. The evaluation team may allow deviations in content, usually due to the TOE being evaluated against one of the TCSEC interpretation documents [14-16].

The description below specifically identifies documentation the evaluation team uses. Where possible, the evaluation team seeks to rely on vendor documentation, rather than on interaction with vendor personnel (who could leave the vendor or be reassigned to other projects), to develop the necessary level of understanding of the TOE. However, some amount of interaction between the evaluation team and the vendor staff is vital to allow the team to resolve interpretation issues and to clear up confusion about ambiguities or imprecision in the vendor documentation. One supporting goal of the evaluation process in general and of the design analysis portion of evaluation in particular is for the evaluation team and the vendor to come to a common understanding of how the TOE's design satisfies the TCSEC C2 requirements.

At a conceptual level, three stages of design analysis activities can be identified. First, the individual members of the evaluation team become familiar with the TOE. Second, they develop a high-level understanding and perform a high-level analysis of the TOE's TCB architecture; based on this architectural analysis, the evaluation team can then plan the detailed design analysis. Finally, during the detailed design analysis stage, individual team members analyze parts of the TCB design in detail, document their understanding in the relevant sections of the evaluation report, review their teammates' analyses of other parts, and perform the TCSEC requirements analysis. The level of effort associated with each stage depends on the TOE's maturity, complexity, and quality of documentation.

3: Gain Initial Familiarity

During this stage, the evaluation team's goal is to become familiar with the TOE. The desired level of familiarity is roughly equivalent to what a system integrator needs to decide whether the TOE is a good candidate for integration into a system and to what an accreditor needs to decide whether the TOE can reasonably be used to enforce enterprise- or mission-specific security policies. That is, the evaluation team seeks to understand the TOE's intended use as a general-purpose product.

Intended use can include the expected operational environment, administrative environment, and configurations (hardware and software). The evaluation team can gain an understanding of these aspects of use by reading marketing literature and user and administrator documentation. In surveying the marketing literature, the team should verify that the configuration(s) in which the TOE is to be evaluated are realistic, in that the vendor presents those configurations as solutions to consumer needs. If it appears that the range of configurations the vendor has proposed for evaluation are inconsistent with consumer needs so that the TOE is unlikely to be used in an evaluated configuration, the team should raise this as an issue to the vendor.

The expected operational environment includes system loading (the ranges of number of users and performance characteristics, which are usually configuration-specific), interfaces, the functionality provided to users, and the security policies the TOE can support. The evaluation team gets a sense of the user interface (graphic user interface and/or command line), identifies network or communications interfaces (protocol suites and hardware interfaces) included in the TOE, and gets a cursory sense of programming, application, and administrator interfaces.

At this stage, the team is interested in identifying which compilers and commercial off-the-shelf (COTS) applications are well supported by the TOE so that team members can recall their personal experiences with those products and get in touch with system integrators who have relevant experience. Some aspects of the TOE's functionality and the possibly of the TOE's security architecture or implementation may be shaped by the applications it is expected to support.

The evaluation team can get a sense of user-visible functionality and the user interface by skimming user manuals. The team can get a rudimentary sense of the supported security policies from marketing literature, can get a better sense from skimming administrator manuals, and should be able to get a clear and detailed understanding from the Philosophy of Protection (POP).

The user and administrator documents allow the team to get an impression of the TCB boundary and interface and to become familiar with the objects and other resources the TCB protects. The team can usually skip or skim large portions of the user documentation which do not deal with TCB functionality. The team's reading of those parts of the user documentation that de-

scribe TCB interfaces can focus on understanding the purpose of each interface, its security implications, and whether the documentation provides adequate information on error conditions and auditing; the team does not yet need to understand how to write programs that call the TCB from the interface. The team reads the administrator documents more closely since many of the administrator interfaces have direct security implications.

During this stage, the evaluation team also identifies historical precedents and assesses their applicability to this evaluation. Team members read evaluation reports from evaluations of similar products to get a sense of the organization and level of detail they will be expected to provide. (Note that this can be risky, in that bad examples could be propagated. Guidance from the TRB or from experienced evaluators can focus the team on good examples.) They peruse the interpretations forum on DOCKMASTER to identify issues they might face. Team members may also try to learn something of the TOE's history as a source of early insight into its design and so they can relate the TOE to other knowledge they have.

In particular, team members should investigate known vulnerabilities in related products. While this investigation begins during the stage of developing initial familiarity, it is ongoing throughout the evaluation. For example, if the TOE is a release of a mature product, team members can look for Computer Emergency Response Team (CERT) advisories [17] and trade press articles on flaws in earlier releases. If the TOE is UNIX-based, team members can retrieve information on common UNIX vulnerabilities. If the TOE is a proprietary system, they can check the computer security literature for lessons learned from attempts to integrate earlier versions into larger systems. Since one of the goals of design analysis is for the evaluation team to explain the TOE's TCB to the TRB, the team is well served by a search of the open literature. The team does not want to find itself unable to answer a question such as "I read about a flaw in that processor in *Computer Week* sometime in the past year or so - how has the vendor addressed that?"

4: Develop High-Level Understanding

During this stage, the evaluation team's goal is to develop an understanding of the TOE's philosophy of protection, TCB architecture, and TCB interfaces sufficient to define team member responsibilities for the detailed design analysis. More specifically, the team must (1) understand the vendor's identification of subjects and objects under the TCB's control; (2) assess the realism and internal consistency of the security policies the TCB enforces; (3) make an initial assessment of the adequacy of the TOE's policy enforcement and supporting mechanisms; (4) develop a basic understanding of the TOE's security architecture and assess the completeness of the vendor's identification of TCB components; and (5) assess the plausibility of the vendor's claims of meeting all the TCSEC requirements.

These activities are described separately for ease of exposition. It must be stressed that they are performed in parallel and with considerable interdependence, based on vendor training and on careful reading and analysis of vendor-developed documentation. As a general rule, the evaluation team reads the POP carefully, then the ASD, and then the ISD, concurrently consulting user and administrator manuals and previously read documents.

4.1: Identify Subjects and Objects

The goal of this activity is to verify that the vendor has completely and accurately identified the subjects, objects, and other resources under the TCB's control. Complete identification means that the vendor has identified everything the TCB controls (including public objects), has characterized those resources outside the TCB's control, and has provided a convincing rationale for excluding those resources, given the policies the TOE is intended to support. Accurate identification means that the vendor has correctly and completely identified the security-relevant information the TCB maintains for each resource.

The evaluation team carefully reads the section on TCB-protected resources in the POP. The team compares their initial understanding of what the TCB protects, based on marketing literature, user documentation, and system administrator documentation, with the detailed description given in the POP. If the team finds any apparent discrepancies, they first read the user and system administrator manuals more closely; if this does not resolve the team's questions, the team asks the vendor to provide clarification.

The evaluation team reads the ASD and the ISD to become familiar with the TCB interfaces. The team may ask for a description of the vendor's approach to identifying subjects and objects. The team then performs a systematic analysis, based on the POP, ASD, and ISD to identify (or to confirm the vendor's identification of) resources under the TCB's control. Many such resources can be identified directly from TCB interface parameters. Such resources include hardware devices (e.g., disk drives, tape drives, CD-ROM drives, VDUs, input devices, caches, memory, peripheral controllers), storage abstractions (e.g., files, directories, windows), communications abstractions (e.g., ports, sockets, messages, semaphores, pipes, remote procedure calls), and processing abstractions (e.g., processes, programs, tasks, jobs, threads). See [9] for a description of a structured approach to

identifying objects. When identifying subjects, the team seeks to verify that all security-relevant attributes associated with each type of subject are identified and that TCB isolation can be enforced.

4.2: Analyze Security Policies

The goal of this activity is to verify that the security policies the TOE supports are realistic and internally consistent. Realistic policies can be applied in the intended operational environments (as described in marketing literature and user documentation), apply to the resources typical customer organizations want to control, and can be administered correctly by an administrator who has received the vendor's administrator training and understands the administrator documentation. In general, a vendor who develops a C2-targeted TOE is strongly focused on user needs and understands what policies are reasonable.

The primary source of information is the POP. The Trusted Facility Manual (TFM) and Security Features Users Guide (SFUG) are supplementary sources of information. Using the guidance in [12], section 4.2.1, the evaluation team seeks to answer such questions as:

- What access control policies does the TOE enforce?
 - What types of access are controlled? Does the TFM describe how to manage public objects or shared resources? On what basis are access decisions made (e.g., user identity, user role, user membership in a group, subject privilege, user privilege, object type), and how does this vary among different types of objects or other resources? How is revocation of access permissions handled? What limitations apply to propagation of access rights? What default access permissions are established, who can change them, and how? Does the SFUG describe how to manage resources in accordance with the access control policy? If an object is only partially under the TCB's control, why? Does the TFM identify any concerns with respect to management of such objects? What is the policy on protecting access control data (e.g., group membership, role membership, privilege assignments)?
- What accountability policies does the TOE enforce?
 - What types of events are auditable? What information is recorded about security-relevant events? Under what conditions can audit data be lost? Why is this justifiable? What is the policy for who can set audit parameters, who can retrieve audit data, who can archive or recover audit data, and how should these actions be performed?
- What system access policies does the TOE enforce?
 - For interactive processing, what constitutes a successful logon? What checks are made as part of the logon process (e.g., time or location of system entry)? For delayed execution, what constitutes a successful submission of a job? What checks are made? Are the checks made when the request for delayed execution is submitted or when execution begins? Once a user has established a session, what changes can be made to the session security attributes? Do different access policies apply based on the device from which the system is entered? In particular, is there a distinct policy for access from an operator or administrator console? What is the policy for protecting and managing authentication and system entry control data? Who is authorized to establish and maintain such data? What is the policy for ensuring that identification and authentication (I&A) information cannot easily be guessed? For example, if a password mechanism is used, are there controls on password aging? Are passwords automatically generated or user-created? If passwords are user-created, are there controls on password syntax and/or dictionary checks?
- What hardware and software privileges does the TOE provide that could be used (or misused) to violate one or more of the TOE's security policies?
- What physical protections are assumed for the TOE's components (e.g., the system console, peripherals, communications media)?

4.3: Assess the Policy Enforcement and Supporting Mechanisms

The goals of this activity are to identify the policy enforcement mechanisms and supporting mechanisms and to verify that these mechanisms, if implemented correctly, enforce the stated policies.

The primary source of information is the POP. The ASD and other design documentation are supplementary sources. The evaluation team seeks to answer such questions as:

- What are the hardware mechanisms used by the TCB to enforce TCB isolation and/or to ensure that the TCB cannot be circumvented?
- What software mechanisms, if any, are used by the TCB to enforce TCB isolation and/or to ensure that the TCB cannot be circumvented?
- What mechanisms are used to enforce the access control policy? Are there multiple mechanisms, depending on the type of resource to which the policy applies? What data structures are used to enforce the access control policy, and how are

those data structures protected? What events related to attempted and successful access to TCB-protected resources are audited?

- What mechanisms are used to enforce the accountability policy?
- What are the mechanisms that support audit selectivity and analysis?
- What mechanisms are used to enforce the system access policy? Does the TOE support multiple I&A mechanisms? What data structures are used to enforce the system access policy, and how are those data structures protected? What events related to attempted and successful system access are audited?
- What supporting mechanisms are provided, and what are the dependencies among policy enforcement and supporting mechanisms? In particular, what object reuse mechanism or mechanisms are provided? If multiple object reuse mechanisms are provided, which mechanisms apply to which classes or types of objects?
- What system integrity mechanisms are provided to verify that the hardware and firmware upon which the TCB relies are functioning correctly?

4.4: Analyze the Security Architecture

The goal of this activity is to verify that the vendor has clearly defined the TOE's security architecture and that this architecture forms a sound basis for the detailed design and implementation of the TCB. A clear definition includes an unambiguous identification of the TCB boundary and a description of the TCB's overall structure. The description of the overall structure usually includes identification of the subsystems (hardware and software) that together make up the TOE, identification of components (or sub-subsystems) of those subsystems, and functional dependencies among subsystems or components.

Historically, getting a clear description of the TOE's security architecture has been a source of difficulty for evaluation teams. This note describes lessons learned from historical experience. Many of these lessons will not apply if the vendor follows [12] and prepares an ASD. However, if the evaluation team accepts a "virtual ASD" (a vendor document providing a road map from the ASD outline in [12] to vendor documentation), some of these lessons will continue to apply.

The vendor staff may be so immersed in the details of their systems that they have lost sight of the basic principles of their design. If an original architect can be found or if some early papers can be found on the concepts behind the system, it may be possible to obtain the information. Many systems submitted for evaluation have no single architect; they have been developed over a long period of time and by different groups of people so that the overriding design may not be found. For UNIX systems, some help comes from the famous text [18] but is of limited use because so many modifications are usually made by vendors to take advantage of hardware advances. In the absence of an ASD, the evaluation team typically spends considerable effort in trying to find the high-level views of the security architecture. In some cases, this means using social engineering to try to get into contact with the vendor's pioneers. In others, evaluators have badgered the vendor contact person until some earlier documents can be found. Failing such help, the team is left to try to form this picture based on the more detailed documentation the vendor provides. This often means asking for design details that the vendor had not thought particularly relevant or that might require some effort for the vendor to obtain. The primary sources of information are the ASD and vendor training. The TFM is a supplementary source of information. The evaluation team seeks to answer such questions as:

- What are the hardware features of the TOE? Which of those features are security-relevant and why? If the TOE includes several alternative hardware configurations, are there configuration-specific differences in the hardware security features? If so, does the TFM reflect any impacts of those differences on how the system administrator should configure or use the system security features?
- Which logical subsystems of the TOE are part of the TCB? Which subsystems directly enforce one or more of the TOE's security policies, and on which other subsystems do they depend? Which subsystems support policy enforcement, but only indirectly? Section 2.1.3 of [12] provides typical reasons for including subsystems in the TCB.
- If a logical subsystem or smaller system component has been identified as part of the TCB, is this identification justifiable? Has any process that, in terms of the TOE's security policies, runs in user space been included in the TCB?
- Following the guidance in section 3.4 of [4], does the vendor's identification of TCB components include
 - All the hardware in the evaluated configuration(s)? All software that runs in a protected hardware state and that performs protection-critical as well as traditional operating system functions? All trusted processes, e.g., processes that run in a hardware state that an untrusted process is allowed to use but that are trusted to perform functions that violate one or more of the TOE's security policies? Tools that the system administrator is required to use and all components that provide the interfaces needed for security related administrative actions? For each such process, what privileges are granted to it? All additional software that some portion of the TCB depends on to perform its functions?
- If a logical subsystem is not part of the TCB, does it contain any components which should be included in the TCB? That is, does it contain components which enforce the security policy or contain data used in policy enforcement, possess (or can acquire) software privilege to bypass security mechanisms, can execute privileged hardware instructions, must func-

tion correctly in order to support system security mechanisms, and/or must be used by the system administrator to administer the system in a secure manner?

If so, are those components correctly identified as part of the TCB even though the logical subsystem of which they are a part contains so many untrusted components that the vendor does not want to identify it as part of the TCB?

- Which parts of the TOE manage shared resources (and thus have the potential to cause or allow information flow in violation of the security policy)? What controls apply to shared resources? Does the TFM identify any concerns related to management of shared resources of which the system administrator should be aware?

4.5: Assess TCSEC Requirement Satisfaction

The goal of this activity is to assess the vendor's claims that the TOE satisfies the C2 criteria. The team carefully reads the section of the POP in which the vendor provides a requirement-by-requirement justification of how each requirement is met (or fails to apply to the TOE, if appropriate). The team identifies other vendor documentation (the ASD, ISD, and detailed design documentation; test documentation; and the TFM and SFUG) to which it will turn during the next phase of design analysis for supporting information and clarification.

The evaluation team seeks to answer such questions as why and how the TOE meets this and if it serves as a starting point for developing the corresponding subsection of the evaluation report. If the TOE is related to similar products that have previously been evaluated, is the explanation of how the TOE meets this requirement similar to the corresponding explanations for those products? If not, how and why do the explanations differ? (Legitimate reasons for differences may exist, but the evaluation team must be prepared to explain and possibly defend the differences to the TRB.) What additional information is needed to verify that this explanation is accurate? Where in the vendor's documentation is that information found?

5: Perform Detailed Design Analysis

During this stage, the evaluation team's goal is to understand the TCB's design in sufficient detail that it can write the evaluation report (the Initial Product Assessment Report or IPAR), answer ad hoc questions from the TRB, and perform test-related evaluation activities. While this paper does not address test-related activities, we must note that it is more efficient to perform test coverage analysis concurrently with the detailed design analysis. Historically, this has often not been possible because the vendor's test documentation was not available yet. Under the current process, concurrent performance of design analysis and test coverage analysis will allow the evaluation team to cross-check information.

Because of schedule and team resource limitations, the responsibility for this analysis and for writing specific evaluation report sections must be spread out among the evaluators. Teams use various techniques for this distribution, but generally seem to permit individual team members to have some voice. The broad categories usually used are to make assignments by functional components, by TCB interface groupings, or by TCSEC requirement. (Of these, the least effective is assignment by TCSEC requirement.) Allocation of responsibilities is intended to ensure that there is sufficient overlap ("double coverage") that the entire system is analyzed properly. See [7] for further discussion of allocation of analysis responsibilities.

The team writes and reviews the evaluation report during this stage. Team review of the evaluation report is a key component of the detailed design analysis. During this review, team members are able to cross check their work with that of others and to develop the detailed understanding of the TCB's design needed to present a correct and comprehensible summary to the TRB. Team members each read the entire evaluation report; when they find text that seems at odds with their own understanding of the TOE, they should discuss the point with the team member responsible for that section or read appropriate vendor documentation.

5.1: Analyze the Policy Enforcement and Supporting Mechanisms

The goal of this activity is to verify that the detailed design of the policy enforcement and supporting mechanisms, if implemented correctly, will result in a TCB that can enforce the stated security policies, is tamperproof, and cannot be circumvented in an obvious way. The primary sources of information are the ASD and ISD. For each mechanism, the evaluation team seeks to answer such questions as:

- What data structures and algorithms does the mechanism use?
- How are those data structures protected? How can they be modified?
- Are the algorithms consistent with the security policies described in the POP? Are there any quirks that should be reflected in the SFUG or TFM (e.g., effects of order in an access control list on whether access will be granted or refused), and if so, are they reflected?
- Do the algorithms rely on any data that is not contained in the data structures the vendor has identified?

- Do the algorithms have any side effects on the identified data structures, other data, or system behavior?
- Is use of the mechanism audited? If so, how? If not, why not?

5.2: Analyze TCB Interfaces

The goals of this activity are to verify that the vendor has thoroughly and correctly identified all security-relevant TCB interfaces and to ensure that any flaws or vulnerabilities associated with incorrect use of TCB interfaces are identified and corresponding countermeasures are discussed in the TFM. "Thoroughness" means that all ways to invoke TCB functionality are identified, including interfaces the vendor does not intend to be invoked when the TOE is integrated into a system. "Correctness" means that for each TCB interface, (1) user documentation describes its syntax, semantics, parameters, and the effects visible to the user or invoking process, and (2) detailed design documentation identifies the TCB data structures used by the TCB to respond to the interface's invocation and describes any effects that are not expected to be visible at the TCB interface but that alter the TCB's security state.

The primary sources for this analysis are the ISD, user documentation (e.g., UNIX man pages), detailed design documentation, and the TFM. The vendor's test procedures constitute a secondary source. The evaluation team seeks to answer such questions as:

- For each logical subsystem of the TCB, has the vendor identified all security-relevant interfaces? (Not all TCB interfaces are security-relevant. The vendor has the option, but is not required, to describe TCB interfaces that are not security-relevant. However, the vendor must provide evidence that they are not security-relevant. See [12] for further discussion.)
 - Has the vendor identified all security-relevant user and system administrator commands to the subsystem and any corresponding restrictions on interfaces from which those commands can be entered? Are all system administrator commands identified in the TFM also identified in the ISD and vice versa? Has the vendor identified all security-relevant program interfaces to the subsystem (typically system calls)? In particular, is user documentation aimed at application developers and systems programmers consistent with the ISD? Has the vendor identified all security-relevant network and client/server interfaces (within the scope of the evaluated configuration and security policy)? Has the vendor identified all security-relevant machine/processor interfaces? Are there security-relevant TCB interfaces that are not identified in user documentation but that could be invoked by users or systems programmers? If so, are these interfaces described in the TFM, and does the TFM discuss the associated risks and risk mitigating actions to be taken by the system administrator?
- For each security-relevant TCB interface, is the vendor's description complete and accurate? In general, the evaluation team looks for inconsistencies between user documentation, the ISD, and vendor test procedures.
 - Does the description identify all data structures used by the TCB to determine what action to take? Are all user input parameters identified? Are parameters outside the direct control of users identified? Does the description identify all data structures that the TCB is expected to modify or could potentially modify as a result of a successful invocation? Does the description identify any data structures the TCB could modify as the result of an error in the use of the interface? Does it describe error messages? How does the accountability policy apply to use of the interface? Is use of the interface an auditable event? If so, what information is (or can be) recorded?

5.3: Verify System Self-Protection

The goal of this activity is to verify that, if the implementation is consistent with the information provided in the vendor's design documentation, the TCB will be well protected against tampering.

The primary sources of information are the POP, the ASD, and the vendor's detailed design documentation. Secondary sources include the TFM and vendor training. The evaluation team seeks to answer such questions as:

- What are the hardware mechanisms that support system self-protection and process isolation (e.g., hardware domains)? How must those mechanisms be used in order to provide the required protection?
- Which hardware domain(s) does the TCB use? Does the TOE use all hardware domains? Are some hardware domains not used by the TCB also not available to user processes? If so, how are user processes prevented from making use of those domains?
- What are the software mechanisms that support system self-protection (e.g., DAC on TCB resources, setting hardware protection bits, system call transfer mechanisms)? How must those mechanisms be used in order to provide the required protection?
- What are the software mechanisms that define address spaces and associate them with users? Is each address space (i.e., the set of all addresses available to a process, including registers, physical memory, and virtual memory) under the TCB's

control individually identifiable and capable of being associated with a user? How does the TOE perform context switches?

- How do the system self-protection mechanisms depend on the hardware and software configuration? Does the TFM provide adequate guidance to the system administrator on configuring the system to avoid creating vulnerabilities in these areas?

5.4: Verify Conformance to TCSEC Requirements

The goal of this activity is to verify on a requirement-by-requirement basis that the TOE meets the TCSEC C2 requirements. If the vendor's documentation conforms closely to [12], the POP will include the vendor's summary of how each requirement is satisfied. In this case, the evaluation team relies on the analysis performed as part of the activities described above, together with the information provided in the ASD, ISD, and the vendor's detailed design documentation, to check the correctness of the vendor's summary. Development of the requirements summary section of the evaluation report should be concurrent with other design analysis activities and should not be left for last.

6: Conclusion

Under TPEP, design analysis is a well-defined, well-structured set of activities that lead the evaluation team to an understanding of the TOE that (1) serves as the basis for testing and determination that the TOE conforms to the TCSEC requirements, (2) can be clearly and succinctly presented to the TRB, and (3) is shared with the vendor.

Variations among actual evaluations have historically been due to several sources which have been addressed in the improved TPEP process; these improvements are being incorporated into the emerging TTAP. A major reason was premature entry of a TOE into evaluation; this has been addressed by separating the advice period from evaluation. Differences in the amount and quality of vendor-supplied documentation have been addressed by the F&C Guidelines. The design analysis process described in this paper can serve as a basis for a CEM for TOEs with a level of assurance commensurate with TCSEC C2. In particular, it serves as an input for evaluation assurance level 3 (EAL3) in the CC.

References

1. *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985, DOD 5200.28-STD, Washington, DC: Department of Defense.
2. Abramowitz, B. S. and Connolly, J. L., "The Trust Technology Assessment Program and the Benefits to U.S. Evaluations" in *Proceedings of the Eleventh Annual Computer Security Applications Conference*, 11-15 December 1995.
3. *Common Criteria for Information Technology Security Evaluation*, Version 1.0, January 1996, Common Criteria Editorial Board.
4. Meyer, K. R. and D. B. Baker, *Evaluator Activities during B2 TCSEC Evaluations (Draft)*, The Aerospace Corporation, Los Angeles, CA, 3 April 1995.
5. Arnold, J., *Level of Analysis for TCSEC Class C2 and B1 Product Evaluations (Draft)*, National Computer Security Center, Fort Meade, MD, 8 June 1994.
6. Chizmadia, D., ed., *An Evaluation Manual for TCSEC/C2 TPEP Evaluations*, Version 1.0 (Draft), National Computer Security Center, Fort Meade, MD, 5 January 1995.
7. DOCKMASTER frequently_asked_questions forum, Technical Review Board/Team Leader (TRB/TL) FAQ, National Computer Security Center, Fort Meade, MD, updated monthly or bimonthly.
8. Baker, D. B., *Trusted Computer System Evaluation Management Plan*, Aerospace Report No. TOR-0086(6777-25)-1, The Aerospace Corporation, Los Angeles, CA, 1 October 1985.
9. Faigin, D. P., Donndelinger, J. J., and Jones, J. R., "A Rigorous Approach to Determining Objects" in *Proceedings of the Ninth Annual Computer Security Applications Conference*, 6-10 December 1993.
10. *Trusted Product Evaluation Management Plan*, 7 August 1992, NCSC.
11. *Trusted Product Evaluation Process (TPEP) Process Action Team (PAT) Steering Committee, TPEP Procedures (draft)*, July 1994, NCSC.
12. PAT Guidance Working Group, *Form and Content of Vendor Design Documentation*, National Computer Security Center, Fort Meade, MD, May 1994.
13. PAT Guidance Working Group, *Form and Content of Vendor Test Documentation*, National Computer Security Center, Fort Meade, MD, May 1994.
14. *Trusted Database Management System Interpretation*, April 1991, NCSC-TG-021, NCSC.

15. *Trusted Network Interpretation*, 31 July 1987, NCSC-TG-005 Version 1, NCSC.
16. *Computer Security Subsystem Interpretation*, Version 1, 16 September 1988, NCSC-TG-009, NCSC.
17. Computer Emergency Response Team (CERT) Advisories, updated as needed, available via anonymous ftp from info.cert.org in the `/pub/cert_advisories` directory.
18. Bach, M. J., *The Design of the UNIX Operating System*, Prentice-Hall, Englewood Cliffs, NJ, 1986.

System Security Engineering Capability Maturity Model and Evaluations: Partners Within the Assurance Framework

Written by Charles G. Menk III

Department of Defense

June 1996

Abstract:

Since the inception of the SSE-CMM program in 1993, there have been some misconceptions within the computer security and evaluation communities regarding its intended purpose. Evaluators in particular have expressed strong resistance to this effort due to the perception that the SSE-CMM is intended to replace evaluated assurance with developmental assurance. That has not and never will be the case. The SSE-CMM efforts can greatly enhance government, corporate, developer, user and integrator knowledge of security in general. As such, the efforts of the SSE-CMM development team are intended to provide significantly improved input to system developers (internal assessments) and the higher level assurance activities (e.g. evaluations, certification, accreditation) efforts (third party assessments). To best address the needs of our customers, the efforts of SSE-CMM and other assurance efforts must grow to complement each other. It will take focused effort from the security community and developmental assurance organizations, as well as industry partners to achieve this goal.

Evaluated assurance, provided by programs like the Trusted Product Evaluation Program (TPEP), has become widely accepted throughout the computer security industry. However, as the state of technology has advanced, the current process and methodology used by the evaluation community have been unable to keep pace with the accelerated development cycles of the advanced products that computer-security customers desire. The deficit of security expertise, as well as unclear and at times inadequate guidance and requirements within the industry and from government agencies has lead to the persistent practice among development organizations developing security as an afterthought or add-on to an existing product. Such practices make correcting security flaws that affect the underlying product expensive, difficult, and time-consuming. All of these factors have forced evaluators to carry out duties and activities far beyond the scope of pure evaluations and to take on the roles of trainer, developer, writer, and quality assurance inspector for the various products that they have been evaluating.

Given these sometimes conflicting demands on the evaluation process, it has become problematic if not impossible (in some cases) to expect the current evaluation approach to continue providing all the product security assurance and keep pace with the increasing demands of computer security customers (i.e. they can not produce enough evaluated products to meet the demand). That is where the concept of an Assurance Framework comes in. Each activity within the security arena (e.g. CMMs, ISO9000, Evaluations) brings with it a certain level of assurance. The composite view forms the Assurance Framework in which a customer can pick and choose products to support their mission based on their risk tolerance and product cost. By allowing certain activities, like the CMM efforts, to address specific assurance needs, the strain on the

evaluation community may be alleviated a little thereby allowing evaluators to focus on the high assurance products while the lower assurance products undergo a less rigorous assessment / certification process.

In the form of the SSE-CMM, developmental assurance can accomplish many needed improvements in the way that INFOSEC products and systems are produced. These improvements may well have a direct impact on the quality of the product's security development and can assist vendors by better preparing their teams for an evaluation. At the higher maturity levels, some of the work now required of evaluators for low assurance products, such as IV&V functions and general security knowledge, can be accomplished during the initial product development. This will allow evaluators to concentrate more of their efforts on evaluation activities and less on security education and or product development for the vendors. The SSE-CMM is a metric for an organization's capability to develop a secure system. Wouldn't it be nice to know an organization has the capability to build secure systems prior to accepting them into a rigorous evaluation activity?

At times, evaluation efforts seem to drag due to evaluator fears of missing critical security aspects and an overall miscommunication between the evaluation team and the developers. The evaluators' fears sometimes lead to "criteria creep" which is the desire to make a product the best it can be, within the criteria, instead of focusing on the requirements themselves and addressing the customers' needs. Miscommunication from the developers tend to result from an inability to clearly describe their product in security terms adequate to suffice the evaluation teams.

The SSE-CMM can help address both of these concerns. If evaluators know that a product is developed in a mature and security knowledgeable environment, they may be more at ease and willing to accept the developers inputs than they have been in the past. The developers, now aware and able to produce the evidence needed, may not feel that the evaluation process is as antagonistic and the communication lines will remain open, thereby enabling a smoother process and hopefully a more expeditious evaluation effort.

Using the SSE-CMM:

The ongoing development of the System Security Engineering CMM (SSE-CMM) has provided a new opportunity to revitalize government efforts to evaluate computer security systems on a timely basis. The understanding and application of the practices and principles embodied in the this model can help address many of the inefficiencies of the current post-development evaluation process. Below is a listing of the current Process Areas (PAs) of the SSE-CMM:

PA Number	Title
PA01	Specify Security Needs
PA02	Provide Security Input
PA03	Verify and Validate Security
PA04	Attack Security
PA05	Assess Operational Security Risk
PA06	Build Assurance Arguments
PA07	Monitor System Security Posture
PA08	Administer Security Controls
PA09	Coordinate Security
PA10	Determine Security Vulnerability

The SSE-CMM can provide valuable insight in the following areas:

Engineering: A security engineering process improvement mechanism

Acquisition: A security engineering capability metric

Evaluation: A capability-based assurance mechanism

The SSE-CMM is designed as a tool for customers, developers and acquisition agencies to use in addressing their specific security needs. This paper will focus on the relationship between the SSE-CMM and evaluations. The three areas of concern for evaluations that we will highlight are System Planning, Development / Description and Testing.

System Planning:

There are two critical security issues that must be addressed early in the product life cycle to set the stage for the development of a Trusted product: policy and model definition and requirement analysis and definition. Historically, TPEP vendors provide policies derived after-the-fact. This leads to inconsistencies as they try to retro-fit the Policy and model to the operational system instead of building the system to address the policy and model specifications. The requirement analysis and implementation is often also done after-the-fact, resulting in last minute additions and modifications that may or may not be consistent with the policy and model. The result is a "dynamic" policy and model that may or may not be able to be evaluated due to its unstable nature.

Developing an accurate security policy and model for a system requires a well defined understanding of security policies and models, as well as a comprehensive knowledge of the system. Most weaknesses in this area result from an inadequate understanding of security modeling or a lack of commitment to security principles. Since the security model forms the foundation for the systems security, any problems with the model can cause cascading delays with the rest of the evaluation schedule while the vendor tries to correct the model and its implementation. Also, as more and more vendors hire third party contractors to write sections of code or purchase large portions (e.g. X Windows) of their system code, the developers are becoming less able to describe the system capabilities without third party intervention. This results in a disconnect between the policy, model and implementation that can not be readily addressed, causing further delays in the evaluations as the vendors attempt to elicit the information from the appropriate code and system developers and relay it back to the evaluators.

When security is done as an add-on, after development, there is a tendency to try to adapt and/or interpret the requirements to fit the product's capabilities. This is usually the result of poorly defined security requirements that allow ambiguity and is sometimes driven by a wholly understandable self-interest on the part of the vendor (e.g. ROI decision process). This can cause recurring delays throughout an evaluation as evaluators have to argue and explain why the vendor's interpretation and/or implementation of the requirement is not adequate to meet the evaluation criteria. The requirement analysis is further complicated when the basis for the analysis is an inadequate policy and model. In these situations, even a correct analysis would yield a failure to address the needs of the customer.

The SSE-CMM can greatly enhance a vendor's capability to address evaluation concerns by interjecting correct order in the system development life cycle. The Specify Security Needs (PA01) Process Area focuses on an organization's ability to capture the customer needs, identify applicable policies, laws, and constraints and identify the appropriate operating environment for the given system. Adherence to this and other process areas will enhance the developers' ability to define a system that addresses all the operational and functional requirements within the desired security specifications.

As part of the SSE-CMM, organizational processes that plan and track changes within the system are implemented. This capability meshes nicely with the evaluation Rating Maintenance Program (RAMP) process, which requires the tracking of security relevant changes to maintain the current rating for the product. Through the adherence to the SSE-CMM tracking requirements, a developer should be able to clearly articulate the changes and provide adequate evidence to support and expedite RAMP activity. This approach is preferable to the after market, add-hock patch-and-play approach used in many systems today because it gives a deterministic procedure for trouble shooting and analyzing the system to address the evaluation issues associated with maintaining a rating.

System Development

When vendors come to an evaluation process, they are expected to bring a complete Commercial Off The Shelf (COTS) system. The intent is that the evaluation group will receive a complete product, design documents and test suite from which they can verify the vendor claims with regard to the level of trust the product will operate under. Undocumented code makes design review, testing and verification efforts more difficult and in some cases untenable.

Malicious Insertion and Inadvertent Errors

The major concern about code within a secure system is whether it does exactly what it claims to do... no more and no less! Undocumented code that is inserted maliciously can lead to the most egregious security violations by adding hidden functionality to the system (e.g. trap doors, trojan horses, covert channels). Although supervisory oversight, peer reviews and shared knowledge of the product development can not prevent all violations, it will tend to reduce the likelihood of successful insertion and in many cases serves as an effective deterrent to such attempts. In addition to malicious attacks, there is always the potential for inadvertent errors. The effects of coding errors are most dangerous in an environment where they are tolerated or undetected. Sloppy development practices and poor reviews contribute to that environment and leave a system vulnerable to the effects of unintended activity. The possibility of hidden functionality and the potential for coding errors increases the time that an evaluator must spend analyzing and testing the system to ensure that it is secure.

The SSE-CMM can play a major role in preventing the likelihood of malicious insertions and inadvertent errors. The Provide Security Input (PA02) and Assess Security Risk (PA05) Process Areas measure an organization's ability to translate the customer needs into appropriate mechanisms within the system and monitor the development effort to ensure the project stays within the risk tolerances. Also, as an organization moves up the SSE-CMM scale, its capability

to conduct effective system and code reviews increases, the accountability is more structured and an effective peer review process is in place. These types of activities act as a strong deterrent to the malicious coder and can reduce inadvertent errors in the early stages of development thereby reducing the likelihood of failures being identified in testing and evaluations.

Design Documentation

Many security concerns are either not addressed in the standard documentation or are discussed in a vague, roundabout way. This produces delays in any evaluation effort because the existing documents must be revised or new documents developed to provide evaluators with the information that they need. These delays have a compounding effect because they occur after the evaluators and developers have already spent a significant amount of review time to determine the document's shortcomings.

Historically, the development and maintenance of documentation has been one of the toughest areas to keep current within a specific product development. This is mainly because developers are not tasked to write the supporting documents and may not communicate directly with the authors, which sometimes leads to confusion among the document writers resulting in contradictory or incorrect reflections of the system. When evaluators review this type of documentation, a significant amount of time is spent talking with the vendor to resolve the contradictions and correct the documents.

All of the issues discussed above with regard to documentation development and review lend to the high preponderance of late delivery of adequate security design documents to the evaluation body. In many cases the necessary information is not readily available to those responsible for describing the system specifications in written format. As a result, evaluation groups are faced with ongoing discussion of whether to believe the documents or the current system code. The question of whether the documents reflect what should be or are incorrect plagues evaluation groups and causes increased concern over the system's security posture.

Within the SSE-CMM, there is significant focus on the documentation produced throughout the System Security Engineering effort. These activities are encapsulated in the Assure Security (PA06) Process Area. As an organization's System Security Engineering capability maturity increases, the documents produced should provide a more clear and concise view of the security within the products or systems under development. This in turn will help evaluation efforts by reducing the number of document review iterations.

System Testing

Many products / systems today provide security as an afterthought, resulting in security-relevant interfaces and components being identified based on a review of the finished product. As such, the probability of missing a security-relevant item is greatly increased. This is often exacerbated by a disconnect between what vendors state as security relevant and what evaluators believe is security relevant. The result is added debate and extended schedules as the two "opposing" views are

worked through to a mutually acceptable compromise. With this approach, the security in any given product / system could be substantially different for each release as the results of negotiations become more subjective due to team composition and time constraints.

It has become more apparent that test suites are not complete when the evaluators are ready to test the system. The testing required for security evaluations is often beyond the scope of normal vendor testing. Therefore the appropriate tests must be created after the vendor decides to undergo an evaluation. Time has shown that vendors are capable of addressing their code, but testing breaks down when third party code, with little or no adequate design documentation or comment, is embedded in the product / system being evaluated. Incomplete test suites are fast becoming a recurring problem that leads to schedule slips that in some cases have placed entire evaluations on hold and in the worst cases resulted in evaluation termination.

In the rush to complete a security test suite, vendors who lack good development practices and quality controls may prepare tests that are not fully functional and in some cases, the output gives little indication of what the test actually accomplished (such as, "Test Passed"). This can force the evaluators to perform lengthy and tedious code reviews. In addition to wasting time in needless code reviews, other evaluation resources have been squandered when evaluators have been called to travel to a test site only to be told that they cannot do their job because the tests will not run. In some cases this failure has placed undue burden on evaluators to develop the right "vendor" tests (a function outside the scope of evaluations). Such failures stem from a lack of understanding the effects of security relevant changes to the base system. Many times vendors assume that a generic test suite will be sufficient to test a modified security platform and therefore no test verification is done, even though this should be a fundamental activity in the development cycle. Once again, we see the effects of reduced time lines and funding taking its toll on proper procedures. Had there been a mechanism in place to track the system security changes and their effects on the original tests, most vendors would be able to incorporate the appropriate fixes within their test suites prior to the actual testing cycle.

The SSE-CMM directly addresses the testing issues through the Verify and Validate (PA03) and Attack Security (PA04) Process Areas. Throughout the development effort, the SSE-CMM places great emphasis on communication between the various system engineering activities and the security activities. This should foster an environment more capable of integrating the security features into the main-line product. The result should be a well defined security / system interface and greater ability to develop complete and accurate test suites.

Conclusion:

Due to the rapidly changing, fast paced system engineering development cycles and increased demand for security, the customer demand for assurance, at a reasonable cost in time and money must be addressed. The SSE-CMM may not be the only solution, but it is a viable one.

Organizations that adhere to a documented, supported and mature security engineering process should be better able to define, build and deliver secure products on time and within budget. In addition to meeting the immediate needs in terms of timeliness and cost, an adherence to SSE-CMM practices should also provide a firm foundation for further, more in depth analysis (e.g. evaluations, accreditation and certification).

Now is the time to begin evolving evaluations and alternative options to ensure Commercial Off The Shelf (COTS) products that are state-of-the-art and secure. Since the RAMP program is so closely related to the CMM efforts, perhaps that relationship may provide fertile ground for the first linkages between developmental assurance and evaluations. The future of secure COTS solutions remains uncertain, but promising, if all the players (e.g. Government, Commercial, Evaluation and Alternate Assurance experts -- CMMs, ISO9000) are willing to work toward a common goal.

References:

SSE-CMM (Draft): TBD. Contact Charles G. Menk III for information: (410) 859-6091, menk@romulus.ncsc.mil.

SE-CMM: A Systems Engineering Capability Maturity Model (sm.), Version 1.1, November 1995, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213

SW-CMM: The Capability Maturity Model for Software, Version 1.1, February 1993, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213

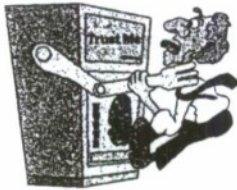
TCSEC: Trusted Computer System Evaluation Criteria, DoD 5200.28-STD, December 1985, Department of Defense.

TNI: Trusted Network Interpretation, NCSC-TG-005, July 1987, National Computer Security Center, 9800 Savage Road, Fort Meade, MD 20755

TDI: Trusted Database Interpretation, NCSC-TG-021, Version-1, April 1991, National Computer Security Center, 9800 Savage Road, Fort Meade, MD 20755

RAMP: Rating Maintenance Phase Program Document, NCSC-TG-013, March 1995, National Computer Security Center, 9800 Savage Road, Fort Meade, MD 20755

The SSE-CMM & Evaluations: Partners within the Assurance Framework



Charles G. Menk III
V21, DoD
menk@romulus.ncsc.mil

DoD

6/20/96

1

Overview

- ✦ Introduction
- ✦ The Assurance Framework
- ✦ Vision and goals of SSE CMM
- ✦ Evaluation Concerns
- ✦ SSE CMM Solutions
- ✦ Recommendations
- ✦ Summary



DoD

6/20/96

2

Motivation

- ✦ Increased use of commercial products
- ✦ Commercial development cycles shorter than evaluation time lines
- ✦ Move from risk avoidance to risk management



DoD

6/20/96

3

Areas of Focus

- ✦ Qualification
 - How to believe someone is capable of doing the job
- ✦ Specification
 - What criteria can be used as a basis for an adequate measurement of capability
- ✦ Verification
 - Proof that the job is done correctly

DoD

6/20/96

4

The Assurance Framework

- ✦ Comprehensive integration of assurance
 - Trusted Capability Maturity Model (TCMM)
 - System Security Engineering CMM (SSE CMM)
 - ISO 9000
 - X/Open Branding
 - Testing
 - Evaluation

DoD

6/20/96

5

Statistics on CMM for Software ROI

Measure	Range	Median
Cost of SPI per engineer	\$490-2004	\$1475
Annual productivity gain	9%-67%	35%
Defects discovered pre-test	6%-25%	22%
Reduction in post-release defects	10%-94%	39%
Value returned on each dollar invested	4-8	5

Source: Benefits of CMM-Based Software Process Improvement, SEI, August 1996

DoD

6/20/96

6

System Security Engineering CMM

- ❖ Apply CMM concepts to Security Engineering
- ❖ Process improvement mechanism
- ❖ Capability-based measurement of Security Engineering process
- ❖ Developed "stand-alone" extension of the SEI System Engineering Methods

DoD

6/20/96

7

SSE CMM: A Community Effort

- ❖ Driven by industry-led Working Groups
- ❖ Facilitated by NSA
- ❖ Office of Secretary of Defense supplemented NSA funding
- ❖ Volunteers for pilot assessments in mid-1996
- ❖ Seeking commercial home for model
- ❖ Internet Web Site:
<http://www.ssecmm.ashton.csc.com>

DoD

6/20/96

8

Vision Statement

- ❖ To provide secure solutions NOW
- ❖ To provide the most assurance at the least cost:
 - In Time & Money
 - To the DoD and its Partners
 - Return On Investment
 - Security within the bounds of budget
 - CMM to target "after-market" cost reduction
 - Risk avoidance (patch and play) to risk management
- ❖ To do it with grace

DoD

6/20/96

9

Goal and Objective

- ❖ To have security built-in from day one
- ❖ To KNOW it is being done right
- ❖ To assist in educating developers, those called upon to analyze the products and systems, and their customers

DoD

6/20/96

10

The SSE CMM Process Areas

- ❖ Specify Security Needs
- ❖ Provide Security Input
- ❖ Verify and Validate Security
- ❖ Penetrate Security

DoD

6/20/96

11

The SSE CMM Process Areas (cont.)

- ❖ Assess Security Risk
- ❖ Assure Security
- ❖ Monitor System Security Posture
- ❖ Manage System Security Controls
- ❖ Coordinate Security

DoD

6/20/96

12

Evaluation Concerns



- ❖ Documentation
 - Does not reflect implementation
 - Not detailed enough
- ❖ Code
 - Not verifiable
 - Undocumented
 - Flaws

DoD

6/20/96

13

How Did We Get Here?

- ❖ Add-on Security
- ❖ Reduced Development Time Lines
- ❖ Systems too large and complex
- ❖ Evaluators expected to do too much
 - Educate about security
 - Assist in design modifications
 - Assist in documentation development

DoD

6/20/96

14

SSE-CMM Solutions



- ❖ Address security from day one
- ❖ Create confidence in developer abilities
 - Documentation
 - Configuration Management
 - Error detection and correction
 - Continuous improvement

DoD

6/20/96

15

Current Status of SSE-CMM

- ❖ Draft 1.0 now available
- ❖ Pilots completed in June-October 1996
 - Combined assessments
 - Stand alone assessments
 - Add-on assessments
 - Special case assessments



DoD

6/20/96

16

Pilot Results to Date

DoD

6/20/96

17

Let Them Eat Cake

- ❖ Bakeries should be able to bake cakes
- ❖ Certain bakers are better than others
- ❖ Some bakers have baked cakes
- ❖ Some use a recipe
- ❖ So how do you pick a good cake?
 - Taste it when done (Evaluate)
 - Pick a professional cake baker (CMM)

DoD

6/20/96

18

Recommendation

- ✦ Guide SSE CMM development to provide output that has diverse utility
 - Evaluations
 - Accreditation & Certification
 - Profiles
- ✦ Use SSE CMM to gain assurance that the developer CAN build secure products
- ✦ Potential support to RAMP process NOW

DoD

6/20/96

19

For Additional Information

- ✦ John Adams
 - Chief, V213 Process Engineering
- ✦ Trusted Capability Maturity Model
 - Amy Mastranadi
- ✦ System Security Engineering CMM
 - Chuck Menk

(410) 859-6091

DoD

6/20/96

20

APPLYING THE TCSEC GUIDELINES TO A REAL-TIME EMBEDDED SYSTEM ENVIRONMENT*

Jim Alves-Foss, Deborah Frincke and Gene Saghi
Laboratory for Applied Logic
Department of Computer Science
University of Idaho
Moscow, ID 83844-1010

Abstract

The DoD Trusted Computer System Evaluation Criteria (TCSEC) was developed to provide a common yardstick for evaluating system security, a guide for system developers, and as a procurement standard. Since these guidelines were released, it has become important to consider the security of systems other than the traditional operating systems that influenced the TCSEC. Multilevel data security is required of many advanced, real-time embedded systems. In this paper, we discuss real-time embedded systems such as those found in avionics systems and how the TCSEC requirements may be modified to suit such systems.

Introduction

The DoD Trusted Computer System Evaluation Criteria (TCSEC) [1] was developed to provide a common yardstick for evaluating system security, as a guide for system developers, and as a procurement standard. However, since then it has become important to consider the security of systems other than the traditional operating systems that influenced the TCSEC's development. Real-time systems used in all types of manufacturing lines may require security. As these systems become more and more integrated and as the amount of data sharing increases, security and validation will become increasingly important. Industrial spying is becoming a very serious problem in some industries.

Real-time systems are becoming quite complex, and may handle sensitive tasks. The F-22 avionics system features a high-performance, shared-memory, heterogeneous multiprocessor connected to sensors and instrumentation by high-bandwidth fiber-optic interconnects. The F-22 operating system supports dynamic assignment of tasks to processing assets (data processors and signal processors). The Boeing 777 has over 2.6 million lines of code in the avionics and cabin-entertainment system. The heart of the system, Airplane Information Management System (AIMS) built by

*The research was funded in part from a grant by Texas Instruments.

Honeywell Air Transport Systems Inc., handles flight management, cockpit displays, and central maintenance, and modules share processors, memory system, and operating system. Software for ATMS alone is over 600,000 lines of code [10].

One definition of a secure system requires that it protect the information it stores from unauthorized release or modification. The Multilevel Security Policy (MLS) as described in the TCSEC (for B1 and above) associates security levels with subjects (e.g., program, user) and objects (e.g., data sets, memory), and requires that the contents of objects can only be seen by a subject at its level or lower; that is, information can flow to the same or higher levels but never to lower levels. This mandatory security policy is augmented by a discretionary policy that further restricts information on a need-to-know basis. More abstract and general models of security that avoid the need to consider objects have been formulated by Goguen and Meseguer [1, 5], Fiertag, Levitt and Robinson [2], McCullough [8, 9] and McClean [6, 7]. In these models, the information a subject observes is dependent on the actions of subjects at the same level or lower. That is, the actions of higher-level subjects cannot be observed by lower-level subjects.

In a computer system the burden of security usually falls mostly on the operating system. An operating system that satisfies the MLS policy must enforce access control: it must not permit processes to have access to objects in violation of the security policy. In addition, the operating system itself must not be a channel for the communication of information not in accordance with the security policy. Such unwanted information flow can potentially occur through objects managed by the operating system and shared by more than one subject, or through timed performance of actions on shared resources. The term *covert channel* is often used in referring to such objects. There have been successful attempts to develop systems that implement the MLS policy, mostly for single host/multiple user systems such as mainframes or shared workstations. Regardless of the policy or model used to develop the system, there is the requirement to provide assurance that the implementation satisfies that model. The TCSEC specifies the types of assurance required to meet various levels of security certification. The assurance may consist of informal arguments, test documentation, formal models and descriptions and formal verification.

Background

Distributed and Real-Time Systems

For purposes of this paper, a distributed system is considered to consist of hosts (e.g., control systems, data acquisition systems, data analysis, and user interfaces), servers (e.g., repositories for objects accessible to multiple hosts, such as files, directories, names, data sets, shared memory), and a network through which the hosts and servers communicate. Security is especially important for distributed systems since such systems often have hosts and servers with different security classifications and certification levels, some with no access control (untrusted) and others that are multi-level secure (MLS). This is especially true of an open-system architecture with components supplied from different vendors.

Architectures for MLS distributed systems vary according to the services offered by the system. In a simple case, each host can support a single user or, more generally, several users operating at the same level. Here the burden of assuring security can fall on the network, which can mediate all communication between hosts to ensure only those intended to communicate with each other do so [11]. Indeed, since users are permitted to communicate only through a few well-defined interfaces,

it is easier to show compliance with the security policy for this distributed system than for most common multiuser mainframes. A more general distributed system would support multilevel hosts. Some of these systems permit the sharing of services or hosts across the system, perhaps through process migration. Here, one must prove the hosts secure in addition to requiring trusted interhost services.

The focus of this paper specifically involves distributed real-time embedded reducing the overhead required for maintaining system security. For the purposes of this paper, a real-time system is one which provides mechanisms to ensure that executing system tasks will meet specific performance and deadline criteria. An embedded system is one which is used specifically to monitor and control attached peripherals (in our example, the peripherals are sensor and weapons systems on a fighter aircraft). Such an embedded real-time system has a very limited user interface and can not be considered a general-purpose multi-user system. The addition of security features to such systems is often seen as an added processing burden that is unrealizable in a real-time system (or at least cost prohibitive). The paper identifies those aspects of the TCSEC which are not absolutely necessary for such systems, thereby reducing the overhead required for maintaining system security.

The TCSEC Guidelines

The TCSEC provides metrics against which systems can be evaluated, guidance for system development, and procurement guidelines. These guidelines specify both system properties and assurance requirements. The system properties are specific to a system configuration and are geared for a multi-user general purpose operating system. The assurance requirements are required for any type of system, regardless of operational environment or design.

The TCSEC's four divisions of certification are lettered A to D, where division A is the highest classification and division D is the lowest. Division D, or *minimal protection* systems have been evaluated but fail to meet requirements of a higher classification. Division C systems have provisions for *discretionary access control* (protection under the user's control) and *audit*. Division B systems additionally provide *mandatory access control*, or must implement mandatory restrictions on information flow between different security levels. The more restrictive subdivisions in this division require a formal statement of the security policy, documentation of the system design, testing to assure that the design is consistent with the specification, analysis of covert storage and timing channels, and permit only security relevant code in the reference monitor¹. Finally, Division A or *verified protection* systems have the same functionality as required for B3 systems, and developers must provide additional assurance that the system design correctly reflects the specification and implements the security policy. Assurance is gained through the use of formal design specifications and formal verification of these specifications.

Within each division of certification is a set of requirements, which specify behavior, design and operation of the computing system. Requirements are divided into one of four categories: security policy, accountability, assurance, and documentation. The security policy provides guidelines for the evaluation of discretionary access controls, object reuse, labeling, import and export of labeled objects, and mandatory access controls. Accountability provides guidelines for the evaluation of identification and authentication mechanisms, auditing, and trusted path access. Assurance provides guidelines for the evaluation of the system life cycle including system design, testing, analysis, management and maintenance. Documentation provides guidelines for the evaluation of the system

¹The TCSEC document defines a reference monitor as a system task that manages all references and validates them according to the security policy.

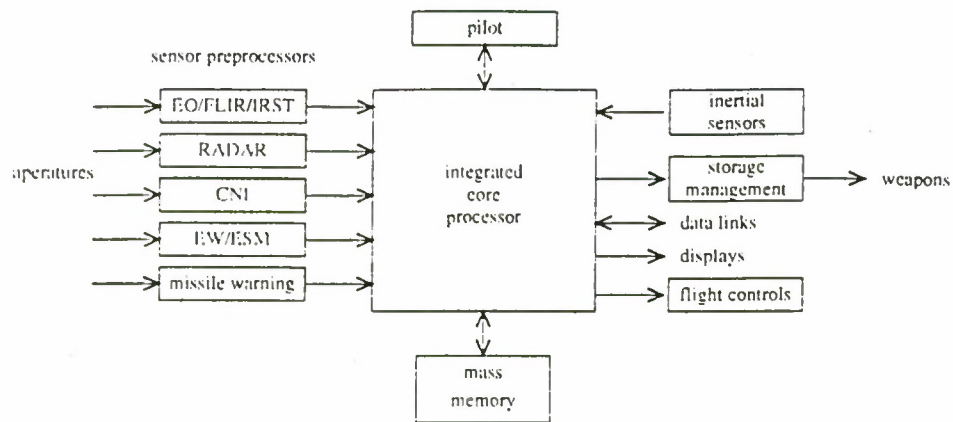


Figure 1: Block diagram of an avionics system.

documentation, both from the user and management perspective as well as from the design and maintenance perspective.

Security Guidelines in a Real-Time Environment

The TCSEC was designed for evaluating multi-user, multi-level (for levels B1 and greater), general-purpose computing systems. Certain real-time embedded systems differ enough from these traditional systems to such an extent that many of the original TCSEC standards do not apply. We present aspects of real-time systems which affect the applicability of the TCSEC guidelines, emphasizing category B3, which incorporates all lower level functionality and differs from A1 only in the amount of assurance required. In the following subsections, we discuss a real-time embedded avionics system, enumerate differences between traditional TCSEC systems and our target real-time system, and analyze aspects of the TCSEC guidelines as they apply to real-time embedded avionics systems. Our discussion is summarized in Table 1.

An Avionics Real-Time Embedded Computer System

Modern real-time computer systems are migrating from purely proprietary architectures to open system architectures. It was once thought that only proprietary architectures could be validated for use in a secure environment, because commercial off-the-shelf components could not be considered trusted. However, Rushby & Randell [11] and Stoneburner & Snow [12] have shown that untrusted, single-level systems can be incorporated into a multi-level distributed system (such as may be found in a real-time computer system), and the result can still be validated. Thus, an open system architecture is a viable alternative to wholly proprietary systems even in a secure computing environment.

A high-level block diagram of an avionics system is shown in Figure 1. An expanded view of the integrated processor is shown in Figure 2a, while Figure 2b depicts one possible implementation of one of the integrated processor's signal processors. To reduce the size and weight of the avionics system, to reduce the number of unique processor designs, and to provide increased fault tolerance, the data processors and signal processors are interchangeable and assigned to tasks dynamically.

In Figure 1, assume that one of the data links operates at a lower security level than the RADAR subsystem. Both may utilize different data processors or signal processors (see Figure 2) at the

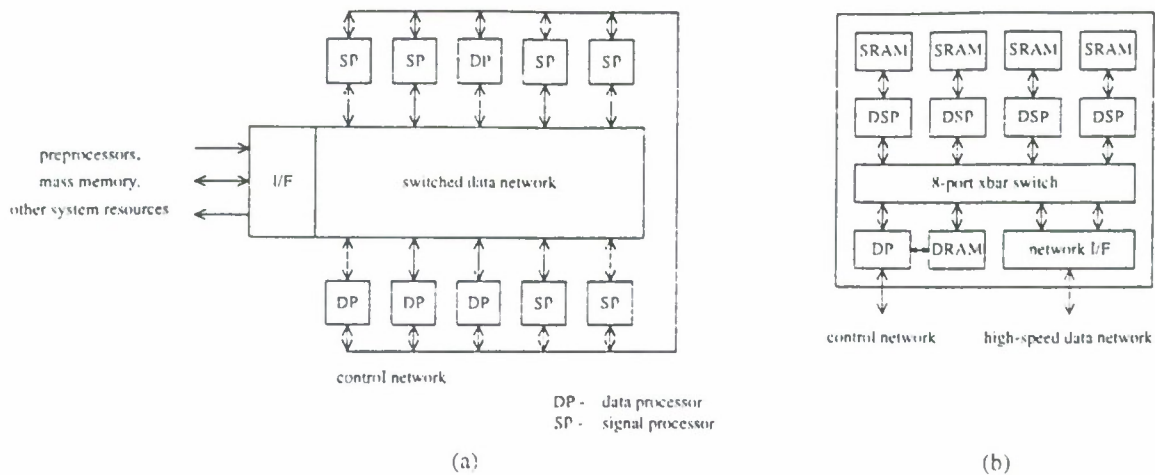


Figure 2: Block diagram of (a) an integrated processor and (b) a signal processor.

same time, passing information across the data and control networks at different security levels. They may even use the same data processors or signal processors at different times. The data processors and signal processors must be capable of operating at various security levels and there must be safeguards built into the hardware and software that prohibit access to secure data by tasks that are not cleared for access to that data, whether that data exists in network messages or in processor memory as the result of a previous task. Potentially, the four digital signal processors (DSPs) shown in Figure 2b operate at different security levels simultaneously.

Important Differences

Traditional systems contain users and user-created objects, including files and processes running on behalf of the user. In contrast, the core processing elements of real-time embedded systems typically consist of a static set of well-defined processes and processing elements. Without such a well-defined set it is hard to ensure the required real-time performance constraints of the system. Similarly, it is unlikely that the users of the system (if any) will dynamically create new processes or objects (such as files). Although real-time systems may involve dynamic execution, resource allocation and scheduling, the possible behaviors of the system are constrained to a well defined set of processes with specific limits on numbers and classes of each created object. A properly maintained embedded system will not have users downloading software from the Internet to execute on the system, or creating new data files with discretionary access controls. Instead, users can often be viewed as direct extensions of the computer system or considered to be separate subsystems executing at a single authorization level. Their interface to the system is through a specific physical interface, accessed through physical authentication controls (a pilot may not supply a password to start the aircraft, but needs to pass an armed guard instead).

Processes should be part of a well-defined static set rather than created dynamically (i.e., not installed or newly compiled by users). Although the run-time behavior of these processes and associated resource allocations may be input driven, normally the relationship between processes and associated protection levels is predetermined and fixed. Thus, information flow and protection controls between processes can be statically defined. This eliminates the need for a discretionary access control policy and permits all access control to be managed by a mandatory policy.

The environment in which a given system operates affects the guidelines which are pertinent to its

evaluation. We make the following assumptions about this environment: (1) Every individual in a position to observe human-readable output (such as that provided by gauges and data records) will be cleared at the appropriate level. (2) The system will not be vulnerable to physical tampering (although may be vulnerable to hardware damage or failure).

TCSEC Guidelines that should be modified

Because of the limited nature of the real-time embedded systems we are examining, some of the TCSEC guidelines are irrelevant or less important.

Discretionary access control We believe that traditional discretionary access control will not be a particularly useful tool in many real-time embedded systems for several reasons:

- Subjects in traditional TCSEC systems generally consist of human users, computers or other mechanical systems, and processes. In a real-time embedded system, we have well-defined processes with well-defined roles. Thus, we expect that if we were to associate sensitivity labels with these processes such labels would never change and would be known in advance.
- Traditional TCSEC systems expect objects to be created dynamically, and the number and ownership/sensitivity of these objects is not expected to be known in advance. This is one of the reasons that discretionary access controls are a useful tool. However, in a real-time embedded system, we can assume that all objects that will ever exist are present or planned at its inception. In addition, the role that each object plays in the system will be known in advance. The major justification for such assumptions lies in the need for precise timing between real-time system components. In order to assure properly timed interactions between components, it will be necessary to know all types of interactions in advance.

Because of the prior knowledge outlined above, we believe that mandatory access controls will be sufficient (and preferable) for defining information flow permissions between subjects and objects, that these relationships will not need to change dynamically, and hence discretionary access controls will be unnecessary.

Identification and Authentication Identification and Authentication will not be important within our real-time embedded systems. Since we are assuming the system is invulnerable to physical tampering, the only processes that exist will be those that are included by the developer. This does, of course, presuppose that, if we are using multi-vendor components, we can trust those components not to introduce intrusive processes. One caveat is that we may need identification and authentication *between* components of real-time systems in some situations. For example, if the avionics system is receiving targeting or mapping information from an external source, it will be important to have some assurance that the incoming information is trustworthy. However, there should not be any intruders in the usual sense.

TCSEC Guidelines that might require modification

Subject Sensitivity Labels If subjects are defined as human users, then subject sensitivity labels may not be necessary, since we are not expecting more than one category of subject to interact with the system during normal operation (the pilot). However, if it would be useful to have categories of subject (such as maintenance, pilot, co-pilot), then subject sensitivity labels will be useful. Also, if processes are considered subjects, these labels will be needed as well.

Trusted Path If we are assuming no human users and no physical breach of the system, then there ought not be any possible tampering of information between components. However, this may not be true if we have untrusted vendor components within the system itself.

Audit Audit should probably be included for purposes of maintenance and performance checks, and possibly for validating the actions of outside vendor components. However, if we do not expect to have distinguishable human users and do not expect intrusive processes, we will not need audit records that are aimed at identifying misuse from these sources. The real-time system must still maintain an audit trail of accesses to the objects it protects. Read access to the audit data is limited by physical means. The variety of events that must be recorded is reduced for a real-time embedded system because of the lack of dynamically created objects.

TCSEC Guidelines that should remain unchanged

Reuse The complex avionics systems now being developed usually rely on reuse of objects such as memory and processors for efficient operation. Thus, criteria providing guidelines for object reuse will still apply.

Labels, Label Integrity, Device Labels, Exportation Since mandatory access controls will probably still be needed, subject and object labels (and label maintenance) will still be needed. If our real-time system includes multi-vendor components (as seems likely), then it will be particularly important to continue to address the issue of labeling devices and passing labeled objects between devices.

Assurance, Documentation These categories of the guidelines are system independent. They provide a mechanism to ensure that reasonable effort was made throughout the system life-cycle to provide correct implementation and operation of the system with respect to the security policy and accountability guidelines. Although the amount of documentation, testing, and verification involved may vary between systems, these requirements must be met for all systems to be evaluated.

Conclusion and Ongoing Work

When compiling a review of TCSEC categories we found that some of the standard TCSEC guidelines were not applicable to a real-time embedded computer system. This discovery led to the review

Table 1: Evaluation Criteria Summary for Real-Time System

Criteria	Appropriate for Real-Time	Comments
Security Policy		
<i>Discretionary Access Control</i>	No	Substitute MAC
<i>Object Rense</i>	Yes	Memory, processors shared.
<i>Labels, Label Integrity</i>	Yes	Needed for MAC.
<i>Exportation of Labeled Information</i>	Yes	Vendor-supplied
<i>Exportation to Multi/Single-level Devices</i>	Yes	components along common bus
<i>Labeling Human Readable Output</i>	Yes	“
<i>Mandatory Access Control</i>	Yes	Flight data recorders, printouts.
<i>Subject Sensitivity Labels</i>	Probably Not	Predefined relationships.
		<i>Subjects</i> are defined
		as <i>human users</i>
<i>Device Labels</i>	Yes	Proprietary and vendor-supplied.
Accountability		
<i>Identification and Authentication</i>	None or Limited	Components untrusted, subjects cannot create processes, hence objects will be statically identifiable
<i>Audit</i>	Limited	Functionality/performance checks, covert channel detection.
<i>Trusted Path</i>	None or Limited	Unnecessary if no human users;
Assurance		
<i>System Architecture, Integrity</i>	Yes	
<i>Security Testing</i>	Yes	
<i>Design Specification and Verification</i>	Yes	
<i>Covert Channel Analysis</i>	Yes	
<i>Trusted Facility, Config Management</i>	Yes	
<i>Trusted Recovery</i>	Yes	
Documentation		
<i>Security Feature User's Guide</i>	Yes	
<i>Trusted Facility Manual</i>	Yes	
<i>Test, Design Documentation</i>	Yes	

of major TCSEC categories and their applicability to real-time embedded computer systems as presented in this paper. In summary this review points out that certain of the TCSEC guidelines, such as discretionary access control, user authentication, and export labels may be trivially satisfied or left unimplemented in a real-time system. Although the claims in the review are generic, we are currently evaluating concrete examples to demonstrate the application of the TCSEC guidelines to specific instances of real-time systems. The work presented in this paper is just the first phase of a larger project that involves the analysis and interpretation of security guidelines for real-time embedded computer systems. We are working on the following related projects:

Formal Specification and Verification of Real-Time Systems. This project involves the use of formal specification and verification techniques for the security analysis of real-time systems. Currently we are investigating the analysis and specification of a real-time embedded avionics control system. The system involves a collection of processing units (potentially off-the-shelf) connected through a shared bus (as depicted in Figures 1 and 2). The paper [3] presents details of a high-level formal specification of the system, including information flow protection.

Formal Mapping of the TCSEC Guidelines to Real-Time Systems. We are currently involved in a project, which is a direct extension of this paper, to provide a mechanism for formally mapping the TCSEC guidelines to real-time systems. Although this project initially involves the avionics system discussed above, we plan to extend that work to other real-time control and manufacturing environments.

References

- [1] Department Of Defense Computer Security Center. *Department of Defense Trusted Computer System Evaluation Criteria*. August 1983.
- [2] R. J. Fiertag, K. Levitt, and L. Robinson. Proving multilevel security of a system design. In *Proc. Symposium on Operating System Principles*, pages 57-95, 1977.
- [3] J. Alves-Foss, G. Saghi, D. Frincke and S. Ghantasala. Multilevel data security for real-time, embedded computer systems: A case study. In *Third AMAST Workshop on Real-Time Systems: Models, Properties and Control*, March 1996.
- [4] J.A. Goguen and J. Meseguer. Security policies and security models. In *Proc. IEEE Symposium on Security and Privacy*, pages 11-20, 1982.
- [5] J.A. Goguen and J. Meseguer. Unwinding and inference control. In *Proc. IEEE Symposium on Security and Privacy*, pages 75-86, 1984.
- [6] J. McClean. Security models and information flow. In *Proc. IEEE Symposium on Security and Privacy*, pages 180-187, 1990.
- [7] J. McClean. A general theory of composition for trace sets closed under selective interleaving functions. In *Proc. IEEE Symposium on Security and Privacy*, pages 79-93, 1994.
- [8] D. McCullough. Foundations of Ulysses: The theory of security. Technical Report RADC-TR-87-222. Odyssey Research Associates, Inc., July 1988.
- [9] D. McCullough. Noninterference and the composability of security properties. In *Proc. IEEE Symposium on Security and Privacy*, pages 177-187, 1988.
- [10] G. Norris. Boeing's seventh wonder. *IEEE Spectrum*, 32(10):20-23, October 1995.
- [11] J. Rushby and B. Randell. A distributed secure system. *IEEE Computer*, 16(7):55-67, 1983.
- [12] G.R. Stoneburner and D.A. Snow. The Boeing MLS LAN: Headed towards an INFOSEC security solution. In *Proceedings of the 12th National Computer Security Conference*, pages 254-266, October 1989.

EDI MOVES FROM THE VAN TO THE INTERNET

Brian Bradford

University of Maryland

Executive Summary:

This paper will give a basic background of the Electronic Data Interchange (EDI) system. Afterwards, it will give examples of EDI systems and what are some of the benefits in using them. After giving a list of examples and their uses, it will discuss how it can be implemented and standardized by organizations that need to transmit information electronically. Then, the paper will discuss the issues based around standardizing the format information sent over the EDI system so that comprehension would be achieved by both the sender and receiver. The body of the paper will discuss the security issues that arise when using an EDI system and what the experts have to say about its own security controls. Finally, the paper concludes with comments about the future of EDI and upcoming security concerns.

Abstract

Electronic Data Interchange (EDI) vendors plan to adjust their business strategies because the Internet provides many of the same features as their value-added network (VAN) services. Surging Internet growth is expected to force companies to stop using VANs for EDI. The companies that continue to offer the service must provide Internet connectivity. Vendors claim the Internet lacks reliability and security whereas VANs have successfully addressed those challenges. Those who support moving VAN services to the Internet counter that authentication and encryption have added stability to the network. When EDI is extended past the VAN onto the Internet, smaller companies are expected to be able to purchase the network services at discounted prices. In this paper, I will discuss the transition EDI has made from using the direct connection with trading partners to the use of the Internet. Along the way, I will discuss some the security issues that are factors in deciding which transmission medium to use.

Introduction

Upon browsing through today's professional automation journals one can expect to encounter several articles about Electronic Data Interchange(EDI). EDI is rapidly becoming a popular subject amongst serious businessmen who realize that our world could look very different if business was done electronically. Some even go as far as to claim that a company that does not start EDI soon, will not make it into the 21st century. Whether or not this may be true, the fact remains that EDI will be of major importance in the business world of the future.

Electronic Data Interchange is a method by which information is transmitted electronically from sender to receiver. Data that would traditionally be conveyed on paper documents are transmitted or communicated electronically according to established rules and formats. Before transmission of data, these rules and formats are agreed upon between the originator and the receiver so that comprehension of the data is accomplished. International and domestic standards are continuously updated to assist in the construction of rules, formats and flow of data. The data that are associated with each type of functional document, such as a purchase order or invoice, are transmitted together as an electronic message. The formatted data may be transmitted from originator to recipient via telecommunications or physically transported on electronic storage media.

EDI typically implies a sequence of messages between two parties (trading partners), for example, buyer and seller, either of whom may serve as originator or recipient. Messages from buyer to seller could include, for example, the data necessary for request for quotation (RFQ), purchase order, receiving advice, and payment advice. Similarly, messages from seller to buyer could include the data for response to RFQ, purchase order acknowledgment, shipping notice, and invoice.

Examples of EDI Applications

Primary applications on EDI are business documents exchanged by trading partners with extensions to government concerns (taxes). Business information encompasses the entire range of information associated with commercial, financial, and industrial transactions. Examples of applications:

- *Vendor search and selection:* price/sales catalogs, bids, proposals, requests for quotations, notices of contract solicitation, debarment data, trading partner profiles.
- *Contract award:* notices of award, purchase orders, purchase order acknowledgments, purchase order changes.
- *Product data:* specifications, manufacturing instructions, reports of test results, safety data.
- *Shipping, Forwarding, and Receiving:* shipping manifests, bills of lading, shipping status reports, receiving reports.
- *Customs:* tariff filings, customs declarations.
- *Payment information:* invoices, remittance advices, payment status inquiries, payment acknowledgments.
- *Inventory control:* stock level reports, resupply requests, warehouse activity reports.
- *Maintenance:* service schedules and activity, warranty data.
- *Tax-related data:* tax information and filings.
- *Insurance-related data:* claims submitted, claims approved.

Benefits of EDI

There are different categories of benefits from any business program. There are the obvious and *tangible* ones that can be achieved by improving the commercial processes. Some of the tangible benefits of using EDI are :

- *Reduction in the transaction cycle time.* It is faster to communicate electronically than by traditional papermeans, especially for trading partners who are large distances apart.
- *Improved accuracy through the removal of rekeying.* If the transaction is received and processed electronically then fewer errors should occur.
- *Lower cost per business transaction.* The cost of creating, handling and processing paper documents can be reduced substantially by electronic communications once the initial development costs have been paid off.

An example of this type of benefit can be seen in the Retail industry, where EDI has been used to ensure that perishable produce on its shelves is as fresh as possible by continuously reducing the supply chain cycle times.

In addition to these obvious benefits, there are the more indirect and sometimes intangible ones. For example:

- Reduced inventory and obsolescence, brought about by the ability to order later, with more accuracy and less forward forecasting, because the order cycle time has been reduced.
- Improved responsiveness of the business because of the increased efficiency of business transactions.
- Higher productivity of staff who do not need to correct errors caused by traditional rekeying of information.
- Enhanced integrity of business information by building auditable electronic business networks.
- Eradication of some of the issues associated with networking between time zones and geographical areas.
- Closer working relationships with trading partners, more trust and hence earlier joint consideration of opportunities and problems, and easier to do business with.
- Exploitation of new business opportunities and markets.

Usually, these benefits are quantifiable even though it may be difficult. For example, the New York Mercantile Exchange's (NYREX) 24-hour electronic data interchange system for energy futures allows business to continue even when the normal trading markets "sleep" and helps overcome the problems of geography and time zones. Basically, in order to prove that benefits have accrued, measuring the process before and after the use of EDI can demonstrate improvements that will develop a tighter control on business processes.

EDI Standards

The American National Standards Institute or ANSI is the coordinator and clearinghouse for national standards in the United States. ANSI does not write national standards, it charters organizations called 'Accredited Standards Committees' or ASCs, composed of voluntary representatives from industry, labor, consumer, and government to prepare consensus standards. Upon public comment and approval, ANSI ASCs publish national standards.(Ref 13)

The ASC X12 (a designation assigned by ANSI) was chartered to develop the structure, format, and content of electronic business transactions conducted through Electronic Data Interchange (EDI). The ASC X12 is administered by the Data Interchange Standards Associations, Inc. (DISA), a not-for-profit corporation. DISA staff manages the ASC X12 membership, balloting, standards development and maintenance, publications, communications with ANSI on behalf of ASC X12, and other duties. The result of the ASC X12 committee's efforts are the ANSI X12 standards.

An EDI transaction involves the electronic transmission of a business document in the form of a Transaction Set, that is prepared in accordance with an ANSI X12 standard format, known as a Transaction Set Standard. The ANSI X12 Transaction Set Standards "facilitate electronic interchange relating to order placement and processing, shipping and receiving information, invoicing, payment, and cash application data." A Transaction Set is the data that is exchanged to convey meaning between Trading Partners engaged in EC/EDI. There are currently 187 Transaction Set Standards published by ANSI X12. (Ref 13)

The Federal Government has endorsed the use of ANSI X12 Standards for EC/EDI with the U.S. Government through Federal Information Processing Standards Publication 161. The DoD has published a set of Implementation Guidelines for an approved subset of the ANSI X12 Transaction Set Standards. Those standards included in the DoD are:

- | | |
|---|--|
| 1. ANSI 824 - Application Advice | 8. ANSI 855 - PO Acknowledgement |
| 2. ANSI 832 - Price/Sales Catalog | 9. ANSI 860 - PO Change |
| 3. ANSI 836 - Contract Award Summary | 10. ANSI 865 - PO Change Acknowledgement |
| 4. ANSI 838 - Trading Partner Profile | 11. ANSI 864 - Text Message |
| 5. ANSI 840 - Request for Quotation (RFQ) | 12. ANSI 869 - Order Status Inquiry |
| 6. ANSI 843 - Request to an RFQ | 13. ANSI 870 - Order Status Report |
| 7. ANSI 850 - Purchase Order (PO) | 14. ANSI 997 - Functional Acknowledgement (Ref 13) |

The ANSI X12 standards were selected because they are already used by private industry. They were developed in compliance with ANSI rules for standards development. They build on the success of private industry in implementing EDI. This means that the program represents a mature technology with immediate savings in time, labor, and resources for both the Federal Government and industry. Using the ANSI X12 standards, trading partners will only be required to support a single hardware & software architecture. This simplifies the training burden for the staff who supervise, operate, and support EC/EDI in their organizations. The EDI end-user staff learn one system rather than many. (Ref 13)

Also, the International standards for EDI is called *EDIFACT* (Electronic Data Interchange For Administration, Commerce and Transport). *EDIFACT* represents the common language for conducting business electronically. International standards for telecommunications were established to give us the means to develop open, electronic, business networks for communication with trading partners around the world.

Value Added Networks

In the past, organizations doing EDI typically have relied on specialized firms called Value Added Networks (VANs) for technical assistance. VANs "add value" to EDI transactions by providing technical support, help desk and troubleshooting for EDI and telecommunications problems. They assist in configuration of software, upgrades to telecommunications connectivity, data and computer security, auditing and tracing of transactions, recovery of lost data, service reliability and availability. Some EDI specific services can include broadcasting an RFQ to a collection of vendors, or storage of EDI information for later search and retrieval. (Ref 14)

Many times, VANs will offer EDI translation capabilities that convert flat text files into EDI X12 or *EDIFACT* format. This translation software may be designed with a particular technical solution in mind. It is important to consider how the software would be used and what applications and telecommunications software would need to interact with it. It could inadvertently lock the organization into using only one supplier. (Ref 14)

EDI Standardization Process

Standardization of message formats, and of data segments and elements within the messages, makes possible the assembling, disassembling, and processing of the messages by computer. Along with these standards, the sole driving force for this rapid development in standardizing business communications has been the need to improve business efficiency and effectiveness.

EDI standards cover the exchange of data relating to security, administration, trading partner information, contracts, and distribution and sales activities. EDI takes a traditional application file, such as an order file, and maps the data into a standard format. The EDI standard is defined by organizations such as ANSI ASC X12. Most North American companies use X12 standards, however, the international *EDIFACT* standards are also used. The process of mapping the order data into a standard file is done using translation software. Once the standard file is created, it is sent to the trading partner via a direct connection or a VAN provider. The receiver uses translation software to unmap the standard order file into a format recognizable by its application programs. (Ref 3)

The most predominant issue with EDI standardization is concerned with standards abuse. EDI standards provide a laundry list of data possibilities to select from for a given transaction set, dictating data requirements and exchange sequence. Rather than use the specific place defined for the exchange of shipping instructions, some users employ another area of the EDI document intended for exchanging other notes and comments. This makes the process of mapping more time consuming, since different trading partners can send the same data in various parts of an EDI transaction set. (Ref 3)

Opportunities On The Internet

Today, thousands of companies are using the Internet. These numbers are expected to increase to hundreds of thousands, possibly one million, by the turn of the century. Companies are using the Internet to pursue business opportunities in three areas: electronic collaboration, information distribution and access, and electronic commerce. (Ref 4)

Use of the Internet for electronic collaboration and information distribution and access has focused interactions among end users and between end users and information sources. The many Internet news groups, file transfer protocol (FTP) archives, and World Wide Web sites are testimony to the continuing and expanding focus on this

type of Internet use. Some business examples of this type of Internet activity include mail communications with customers and business partners, the use of telnet for direct sales, the use of FTP for maintaining public archives and for delivering software patches, and numerous internal and external projects utilizing the World Wide Web.

Electronic Commerce (EC), and, particularly, electronic data exchange use of the Internet has focused on providing company-to-company standards-based, secure business transactions electronically. Collaborative efforts such as CommerceNet and individual business pilots between companies are currently under way to test and broaden the use of the Internet. A key component in these efforts is addressing the issues associated with the application-to-application and end-user-to-application interfaces prevalent in EC/EDI.

The Internet provides a variety of capabilities available for EC/EDI use including mail, file transfer, World Wide Web, and remote log-ins. One of the major issues faced in using the Internet for EC/EDI is how to deal consistently with this variety. While transmission control protocol/Internet protocol (TCP/IP) provides the underlying transport protocol for the Internet, the applications must support different protocols dependent on usage. For example, a business application may need to utilize the simple mail transport protocol (SMTP) for mail, FTP for file transfer, hypertext transfer protocol (HTTP) for World Wide Web access, and telnet for remote log-ins. Each of these application protocols presents different limitations with respect to use and value-added functions such as security, encryption, and non-repudiation.(Ref 4)

Taking mail as an example, SMTP, as defined by the Internet Engineering Task Force (IETF) standard request for comment (RFC) 822, performs the message transmission function, but only supports seven-bit American standard code for information interchange (ASCII) transmissions, limits the number of recipients, and often limits the maximum message size. Modifications to SMTP were needed to address the needs of EC/EDI. These modifications came in the form of the multipurpose Internet mail extensions (MIME). MIME defined mail body part structure and content types that provided an SMTP-compatible way to encapsulate documents in mail messages while supporting multipart content types including text, audio, image, video, and even application data. MIME also provided support for several content-transfer encodings including base 64, which enabled incorporation of eight-bit binary data as seven-bit ASCII data.

Further refinements were introduced in RFC 1767 to specifically address the encapsulation of EDI objects within MIME. This permitted the transmission of EDI transactions through Internet mail supporting both EDIFACT and American National Standards Institute (ANSI) X12 EDI standards as MIME content types and ensured that EDI objects retained their syntax and semantics during transmission. RFC 1767 also established an EDI-consent MIME content type as a catch-all to enable trading partner--specific EDI content types to be defined.(Ref 4)

Internet Security Measures For EDI

With MIME and encapsulation of EDI objects within MIME now in place, the focus has shifted to how best to secure EC/EDI transactions over the Internet. RFC 1767 did not provide any security-related mechanisms, but did acknowledge the need to address authentication, data integrity, privacy/confidentiality/access control, and non-repudiation. It recommended the use of either Internet mail-based security or EDI-based security.(Ref 4)

For Internet mail-based security, two primary approaches have emerged: privacy enhanced mail (PEM) and pretty good privacy (PGP). PEM capabilities are described in RFCs 1421 to 1424, and provide for the confidentiality of messages via encryption, originator authentication, content integrity via message integrity check (MIC) algorithms, and non-repudiation if a public key mechanism is used, PGP, a privately developed public/private key system, provides mechanisms for encryption and authentication.

For EDI-based security, many companies deploy firewalls that selectively restrict mail access to and from the Internet. These security firewalls are capable of monitoring and controlling incoming electronic mail information, hiding the internal network structure from outside access, and encrypting/decrypting and signing/validating messages from outside the firewall. A more recent development, currently being piloted, is the incorporation of security features directly as part of the EDI software.(Ref 4)

Today, companies using Internet mail (SMTP and MIME) to transmit EDI transactions have essentially replaced the EDI transmission components previously provided by value-added network (VAN) suppliers with Internet modules. The basic transaction flow is from the business application or database through the EDI translator to MIME for encapsulation and SMTP for packaging, submission, relay, and delivery. At the other end, there is SMTP and MIME stripping, and then passage through the EDI translator into the receiving business

application or database. Where security is a concern, modules are added either as part of the EDI or Internet flow to address encryption, authentication, and non-repudiation issues at both ends.(Ref 4)

EDI On The World Wide Web

Many of the same issues arise when using the World Wide Web for electronic commerce and EDI. The initial use of Web implementations was to provide company information, including information on products and services, on the Internet for access and viewing by end users. The protocol interface for making this information available is HTTP.(Ref 4)

Most companies that have established "home pages" on the Internet for displaying information using HTTP still rely on telephone, fax, or electronic mail to handle order placement functions. Some companies, however, are experimenting with the use of forms that can be filled out on line and submitted for processing by business applications. In these instances, a common gateway interface is established to provide support for forms processing and to interface the World Wide Web to the business applications.

To make business transactions easier on the World Wide Web, hypertext markup language (HTML) allows the creation of forms and provides a vehicle for passing form information to business applications. In addition to text input, HTML forms also support the use of more sophisticated capabilities as pop-up menus, scrolling lists, check boxes, and submit buttons.

As with mail, security is a real concern for electronic commerce and EDI on the World Wide Web. There are two primary areas of concern: Encryption of business transactions to protect contents such as order information and shipping and billing information from alteration or replacement; Protection of any payment information such as credit card or electronic funds transfer information.

Similar to PEM and PGP for mail, there are two emerging security standards and implementations for the World Wide Web. These are secure hypertext transfer protocol (SHTTP) and secure sockets layer (SSL). SHTTP is an enhancement to HTTP developed by the Web Transaction Security Working Group of the IETF SSL was developed by Netscape, Inc., a provider of one of the more popular World Wide Web browsers. Work is currently under way to move these separate security implementations to a common World Wide Web security and encryption capability.(Ref 4)

Whether you intend to use mail, the World Wide Web, file transfer, or any other capability for electronic commerce or EDI business transactions on the Internet, you should strongly consider implementing a secure gateway to control access to and from the Internet. A security firewall or similar capability is essential to protect your business applications and resources from unwanted access or tampering. You may also want to consider establishing separate "logical domains" to further isolate your applications from Internet-access capabilities such as MIME and HTTP. Until security and encryption capabilities are standardized, it makes sense to separate the current implementations from your applications so, as changes occur, they do not directly impact those applications.(Ref 4)

Examples Of EDI Security

Premenos Corporation, a provider of EDI software for electronic commerce applications, has announced its efforts to co-develop a user agent with VeriSign Company, whose charter is to provide digital-signature certification for document authentication.

Premenos recently announced Templar, an EDI-authentication agent to enable confidentiality, authentication, data integrity, and non-repudiation of both origin and receipt. Certification provided by VeriSign will provide TCP/IP business users the kind of scalability to make possible wide deployment of Templar and EDI over the Internet.

Templar software is a layer between the mail agent, such as SMTP/MIME, Lotus Notes, Microsoft Exchange, or X.400. It is the EDI translation software to ensure confidentiality, integrity, authentication, and non-repudiation of both origin and receipt. Templar software also provides operations management, including trading partner set-up, designation of communication and security requirements, key management, and transaction tracking.(Ref 5)

Templar services include 24/7 customer support, education, and training, and specialized services that include trading-partner implementation, system design and configuration, business process automation, and customized projects involving Templar. Premenos will work with Templar customers to provide firewalls, Internet access,

private TCP/IP network configuration, and value-added network (VAN) gateway services, depending on their electronic commerce strategy needs.(Ref 5)

Templar incorporates RSA's public-key cryptography technology integrated at the application layer--the layer closest to the data. The Templar software agent is completely independent of the mail protocol and the underlying network. Premenos' products are based on industry standards including ANSI, UN/EDIFACT, DES, and SMTP/MIME. Based on a client/server architecture, Templar is written in Object-oriented C++.(Ref 5)

The USPS has also been pilot testing the Templar product from Premenos Cop., Concord, Calif., for secure electronic data interchange over the Internet. Templar incorporates public/private key encryption technology. "The intention is to provide an open EDI opportunity, to conduct EDI without the elaborate trading partner structures and private networks," said Rothwell. As the certification authority, the USPS would register a business's public key.(Ref 1)

The VAN And The Internet

The use of EDI over the Internet is in the early stages, although the technology and services are developing rapidly. In the past, organizations doing EDI relied on VANs for technical assistance. Many of these organizations will look to their VAN for assistance in using the Internet.

VAN services have typically used proprietary network or a networked gateway with a specific set of other proprietary networks. In contrast, an Internet Service Provider (ISP) offers generic network access (i.e. not specific to EDI) for all computers connected to the Internet. A direct internet connection permits real time computer-to-computer communication for client-server applications. Alternatively, a part-time internet connection can be used to access internet servers using an on-demand basis, or access another system via email which includes a store and forward method. Internet email may be used as a gateway to proprietary networks if it has an email gateway.(Ref 14)

Internet email can be configured for a dedicated connection with real-time transfers, or a store and forward method (like traditional VANs), or a combination of the two. For example, this occurs where a direct delivery to a trading partner's system is used when a link is operational, and a store and forward from an ISP is used as a backup.(Ref 14)

A large organization can connect their network to the Internet at an internet exchange point, however, most use a commercial ISP, either a major backbone provider, or local resellers of service off one or more backbones. The ISP provides technical assistance and access to local telecommunications links.

The Internet E-mail standards have hierarchical address spaces that are defined and updated in what the Internet calls "domain name servers." Unfortunately, X12 has a flat address space. So, when an interchange is sent (not via the Internet) to a partner who is on a different VAN, your VAN must do a table look-up to figure out what VAN the receiving party is on. If you use only X12 without the Internet, before you can send a message to this partner, you must first contact the recipient's VAN and have them add you as an entry to his VAN's table. If the ISA contained the VAN ID of the recipient, then you could (in theory) send interchanges to partners via the VAN interconnects without having to notify the recipient's VAN first. However, this theory needs to be worked out in practice. In contrast, thanks to the domain name service, Internet e-mail users (and Postal users) don't have to call up their service provider before sending a message across an "interconnect" to another service provider.(Ref 14)

All VANs connected to the Internet are connected to one another, thus avoiding most of the problems of interconnecting proprietary networks. VANs can then focus on services to their customers such as automatic bid submission, market and business opportunity analysis, and translation software.

EDI Via The Internet Without A Van

In order to use the Internet directly for exchanging EDI messages without going through a VAN, you and your trading partner must agree on one of the Internet protocols for exchanging messages and then agree upon some details with the exchange.(Ref 14)

a) Email based messaging

The simplest and most widely supported means of exchanging messages is via internet email. Typically, the IETF-MIME encapsulation specification would be used to enclose the EDI data within the email message, and the trading partners would need to agree upon an encryption method for secure email, typically PEM or PGP.

The trading partners would then exchange:

1. The internet email address for EDI messages.
2. An internet email address for personal communications related to EDI.
3. Agreement on the encryption and digital signature protocols, including email acknowledgment.
4. Public Keys for PEM or PGP encryption and digital signatures (or private keys for DES encryption).
5. Agreement on the format of the message, e.g. IETF MIME/EDI.

b) FTP based messaging

To exchange EDI messages via FTP, some setup information must be included in the trading partner agreement. Typically, an account would be created for each trading partner for a FTP login, including a password. Usually, each X12 or EDIFACT message would be stored in a file, and the trading partner agreement would define the conventions for naming files and directories for the messages.

The trading partner agreement would include:

1. FTP login name and password.
2. Machine(s) from which the login will be accepted.
3. Additional security protocols, e.g. Kerberos.
4. Directory and file naming conventions
5. File encryption protocols and keys
6. Wrappers around EDI data, e.g. MIME/EDI headers, PEM/PGP wrappers, etc.

There are several compression routines and utilities available for virtually any computer system that uses the Internet. Many of these utilities will convert across platforms (e.g., UNIX to Mac, UNIX to PC, and vise versa) and are available for free from one of several FTP archive servers. Use of these compression routines should be used with care when one is employing an encryption technique such as PEM or PGP.

Example of Connecting Existing EDI Systems To The Internet

Sterling Software Inc. will start shipping a number of products designed to tie existing electronic commerce systems into the Internet. Sterling's Electronic Commerce Gateway, which became available in January, is a suite of software and services that extends the reach of Electronic Data Interchange (EDI) value-added networks to companies via the Internet.(Ref 11)

The Sterling offering includes Dataguard, a client/server encryption product based on the X12 EDI standard. Once files are encrypted, they can be sent to the appropriate user on any EDI VAN or the Internet. To manage encryption keys between users, Dataguard will use Veil, a government created system for which Sterling will become the exclusive commercial licensor.(Ref 11)

Electronic Commerce Gateway also includes a messaging gateway to the Internet off of the Gentran Server. It maps data from application files into EDI format.(Ref 11)

Leveraging its existing EDI VAN, Sterling announced it will provide Internet, X.25, and Systems Network Architecture gateways to Commerce Network, which will enable connectivity to other VANs, such as AT&T EasyLink and GE Information Services EDI Express. As an option, the suite comes with Connect Firewall, Sterling's Internet firewall software for enterprise networks.(Ref 11)

Example Of Moving From VAN To Internet Only

AVEX Electronics Inc., seeking to reduce connection costs, decided to move much of its EDI from a private VAN to the Internet. AVEX generates such typical EDI transactions as purchase orders and invoices; EDI eliminates much faxing, copying, data entry and data processing and lets the company perform more transactions with fewer administrative staff. The current AVEX EDI system runs on an IBM AS/400 with software from Data3 Inc. and allows interaction with more than 50 suppliers and customers. AVEX was able to move to the Internet with the development of Premenos Corp's Templar agent for secure transactions. Convincing trading partners that the system is secure and stable is AVEX's biggest challenge. Tracking EDI packets across Internet segments can be difficult. AVEX has only convinced three suppliers so far to exchange EDI over the Internet.(Ref 8)

The Future Of EDI With Expert Opinions

Electronic Data Interchange vendors and their VANs have thrived for the past 20 years. But with the Internet threatening to overtake their services, many of these vendors have begun reworking their strategies. Analysts predict that in the next five years only a few large companies will continue using VANs for EDI; and those VANs will need to provide Internet connectivity in order to survive.

"Companies will eventually move onto the Internet (for EDI)," says Tim Sloane, an analyst at Aberdeen Group Inc., in Boston. "The issue is not if, but when." (Ref 2) Two of the largest EDI vendors, GE Information Services Inc. and Harbinger Corp., have already announced Internet support. "They feel the threat of the Internet," says Tom Pincince, an analyst with Forrester Research Inc., in Cambridge, Mass. (Ref 2)

Some EDI vendors counter that the Internet currently lacks security and reliability. Analysts, however, say these arguments are becoming hollow. "If I were a VAN, I would try to convince you that there were security and reliability issues with the Internet -- but there aren't," says Pincince. (Ref 2)

Most of the major VAN carriers are either working on or have released similar services already, said Amie Shapiro, an analyst at International Data Corp. "Everyone wants to use the Internet for EDI, and many VANs are addressing their security concerns," Shapiro said. "I think companies will be hesitant to conduct EDI over the Internet, but once it's tested, tried and true, companies will start using the Internet," she said. (Ref 10)

Specifically, technologies such as encryption, authentication, and return receipts are making the Internet a more stable backbone. EDI vendors traditionally have used proprietary software running over VANs to provide turnkey solutions for communities of interest, such as large manufacturers and their suppliers and customers. (Ref 2)

Internet vendors, meanwhile, are promoting standards-based electronic commerce solutions that would enable secure transactions on a casual basis between suppliers and customers. Most Internet commerce product solutions are geared toward individual consumers. But some companies are looking to the Internet as a future platform for EDI transactions as well. Extending EDI beyond the VAN to the Internet would open the market to serve medium-size and small businesses. And greater competition among EDI vendors using Internet connections will probably bring EDI prices down low enough to accomplish that goal. (Ref 2)

EDI may be the application that will introduce the Internet into mainstream IS. EDI, which has existed for two decades, may supply a new information transport path that is markedly less costly than the proprietary networks in use today. However, the transition to EDI on the Internet will require hard work and changes. One expert suggests that adding Internet technology to EDI is a great idea, but the realities are very challenging and complex. (Ref 6) The Internet is much less costly than proprietary networks, but the change will require IS personnel to be EDI administrators. Users, for their part, would have to create or buy their own security products and develop redundant backup systems that would come up whenever Internet transmissions have problems.

Future Transitions To The Internet

User efforts to adopt EDI services should get a boost from products provided by a trio of vendors that move EDI off value-added networks and onto the Internet.

Sterling Software Inc., Harbinger Corp. and Premenos Corp. --significant players in the EDI market--said they are embracing the Internet to provide their large corporate customers with access to a greater number of suppliers and contacts.

EDI generates the bulk of electronic commerce today--\$130 billion of goods were transacted worldwide through EDI versus \$70 million in transactions over the Internet during 1995, according to an October report from Input, a Mountain View, Calif., research company.

Sterling last week announced exclusive licensing of Veil, an encryption and key management software, which is a stand-alone product and as a component of its EC Gateway software and services.

The EC Gateway includes the Gentran:Server, an EDI translator and messaging management product; Connect:Firewall; and Veil and Gentran:Dataguard, encryption for creating a secure tunnel over the Internet to send data. Sterling provides the clearinghouse, or VAN, which clears the EDI transactions.

To automate bill payment, Sterling announced an alliance with Visa International, which creates an end-to-end electronic network between customers and their banks, as well as billing corporations and their banks.

"If the Internet becomes the way to get to more suppliers, especially smaller suppliers, and as we build our global network, it may be easier, more efficient and less expensive to reach out to those in other countries that don't have sophisticated communications techniques like we have," said Roger Trout, EDI manager for Mobil Oil Corp., Fairfax, Va., a user of Sterling's EDI software.

Harbinger, Atlanta, also an EDI VAN service that provides EDI services for Sprint customers, announced its Internet strategy for shipping products in the second quarter of 1996. The company will ship a browser, firewall, gateway and security software called TrustedLink, which supports SMTP and S/MIME. Harbinger's software will let users connect directly over the Internet to other users of the same Trusted-Link software, as well as allow connections over its Internet Value-Added Service.(Ref 7)

Premenos, Concord, Clif., introduced Premenos WebEDI, World Wide Web-based software available in the first half of 1996. WebEDI is targeted at anyone with an Internet connection and a Web browser who wants to perform EDI transactions to encourage more robust worldwide EDI deployment.(Ref 7)

Discussion

There is a tendency for each organization to establish its own rules and administrative policies, leading to rising costs of dealing with multiple trading partners, each intum with its own requirements and procedures. However, new technologies and business practices are necessary if EDI is to move beyond the 30 to 40,000 organizations presently using EDI. According to Department of Labor and Internal Revenue Service statistics, there are about 6.2 million entities with employees and about 14 million other "business" entities.(Ref 14) A business that wants to sell chairs, for example, would have to check with many different customers to see if they had any requirements. By making it possible for a business to use a common method to look for customers, the barriers entering to the electronic marketplace are greatly eased. This does not mean that there is only one source that everyone goes to for a list of current business opportunities. Rather, a prospective supplier only needs to go to a single electronic marketplace. To communicate with each other, the various participants in electronic commerce need to harmonize their procedures and processes. Examples include common trading partner registration and the adoption of standard implementation conventions for EDI messages.

Keeping this in mind, the Internet can be used to send transaction sets to existing trading partners via SMTP or FTP messages. VANs were typically used for *bilateral relationships* between companies, whereas the Internet is useful for establishing *multilateral relationships*.(Ref 15) These bilateral relationships are usually quite stable, but both parties had to agree to share the same VAN or get their VANs to interconnect. Multilateral relationships are between organizations that don't necessarily have existing relationships and may be rather ehpemeral. The Internet is suited to dynamic multilateral relationships that may later evolve into static bilateral relationships between companies using VANs. Therefore, the issues concerning the Internet (security, availability, etc.) are manageable in the early stages of forming a relationship. If your current VAN is not capable of using the Internet, you may need an alternative route for those messages. Later, as the business relationship matures, the use of VANs may be appropriate as the level of communication becomes more important. For example, unless your system has a directory of all registered trading partners, you lack the capabilities to screen and validate transactions that arrive at your site.

Conclusion

We have discussed EDI progress from the VAN to the Internet. Throughout the discussion, we learned that, initially, very few changes may be apparent. New and existing VANs will use the Internet to collect and disseminate EDI transactions; trading partners may be totally unaware of the change in technology. Prices may fall as VANs share telecommunications resources through Internet Protocols rather than maintain their own costly proprietary telecommunications services. Instead of competing with VANs, the ubiquitous connectivity of the Internet offers VANs even greater business opportunities. General purpose Internet Service Providers (ISPs) do not typically offer EDI specific services, but they can provide an alternative means to transfer EDI messages at a small fraction of the cost of typical EDI VANs.

The impact of an organization's moving EDI onto the Internet, independent of a VAN, is more difficult to assess. In the view of some, the introduction of the Internet in the near term (1-5 years) adds additional interfaces and complexity to the organization's existing EDI environment. This may in the short term increase costs and raise new costs. But a corporate commitment to an open systems environment through the use of Internet Protocols offers the potential for a greater interoperability, integration of application systems, and therefore the promise of

higher performance and lower costs. Some organizations will be able to get to these benefits others will pay for a set of largely incompatible services. The return on investment largely depends on one's ability to consider EDI on the Internet as a part of the organization's overall information systems strategy and the organization's plans for a presence on the Internet.

References

1. Bucken, Michael, "CyberPostman rings...." *Software Magazine*. Jan 1996, v16 n1, p.18.
2. Davis, Jessica and Michael Parsons, "EDI vendors adjust strategies in face of growing Internet." *InfoWorld*. Dec 25, 1995, v17 n52, p.39.
3. Jilovec, Nahid, "Making EDI work." *MIDRANGE Systems*. Feb 16, 1996, v9 n2, p36.
4. Muiznieks, Vik, "The Internet and EDI" *Telecommunications*. Nov. 1995, v29 n11, p.45.
5. Muiznieks, Vik, "Premenos promotes EDI security over the Internet. *The OSINetter Newsletter*. July 1995, v10 n7, p.19.
6. Nash, Kim S., "Internet EDI on horizon: users worry that steep do-it-yourself security costs may spoil the view." *Computerworld*. Jan 29, 1996, v30 n5, p.65.
7. Rodriguez, Karen, "New products to boost EDI services, focus on Internet." *CommunicationsWeek*. Dec 4, 1995, n587, p.8.
8. Streeter, April, "Network of dreams." *LAN Times*. Feb 19, 1996, v13 n4, p.86.
9. Schwartz, Jeffrey, "EDI, E-mail transactions made secure." *CommunicationsWeek*. February 12, 1996, n596, p.4.
10. Schwartz, Jeffrey, "PCMCIA card gets smart." *Electronic Engineering Times*. February 12, 1996, n888, p.74.
11. Wingfield, Nick, "Sterling introduces Web commerce tools." *InfoWorld*. Dec 11, 1995, v17 n50, p.66.

On-Line References

13. *EDI Standards*. "ANSI X12 Standards for EDI." April 1995.
14. *Internet-Draft*. "EDI Meets the Internet." April 1995.
15. *Internet-Draft*. "US Federal Involvement." April 1995.

An INTERNATIONAL STANDARD for the LABELING of DIGITAL OBJECTS

(Proposal for a Consumer Protection Act of Digital Products)

Viktor E. Hampel

Technical Advisor on Data Protection to Government and Industry

Hampel Consulting

1515 Jefferson Davis Hwy., Crystal Square Suite # 913

Arlington, VA, 22202-3312, (703) 413-8836

E-Mail: vhampel@attmail.com

Executive Summary

This report proposes collaboration towards an international standard for the commerce of digital objects. In the United States, it should derive its authority from a "*Consumer Protection Act for Digital Products*" patterned after the "*Food and Drug Act of 1906 (21 USC)*". This would bring under control the lack of accountability on the information highways with the help of an agency, similar to the Food and Drug Administration (FDA), mandated "*to develop administrative policy for the safety, effectiveness, and labeling of digital products, and to review and evaluate new applications of such products.*"

It is now technically and economically feasible, and the enabling standards are in place for data authentication, protection, labeling, and safe conduct over open channels: Digital product labels in the structured header of a standard envelope would define the rights and powers of copyrights holders, equivalent to the *visual* copyright notices on title pages of bound literary works and movies. Descriptive digital product labels would advertise their source and content, equivalent to the *visual* content labels on food containers mandated by the FDA. Certified digital signatures would bind this information in the header to the body of the object to assure its integrity, ownership, and use.

Registered Digital Product linked to header with certified digital signatures	Digital Signatures	Header Information with digital labels
--	-----------------------	---

Missing is an agreement on the packaging and description of digital objects based on contents to assure their safe storage, retrieval, transport and use. This forces present electronic mail systems to encapsulate each object either as 'unknown,' or to deliver distinct service for security and processing. Current work underway towards *enveloping and modularity in CALS*, a preferred *Cryptographic Application Program Interface (CAPI)* for cryptographic service API, and a standard *Message Security Protocol (MSP)* should be extended to digital objects in general, mindful of their legal implications for global electronic commerce.

President Clinton challenged us *to move in a New Direction and build Economic growth for America with Technology*. Non-partisan legislation towards this goal would harmonize the related efforts worldwide. Norms for global electronic trade would strengthen the interests of vendors and the rights of consumers. Agreement on a minimum, necessary and sufficient liability and accountability would deter counterfeiting, piracy, and increase trust among business partners. A "*Consumer Protection Act for Digital Products*" would also help to resolve the controversy over the constitutionality of legislation to reduce violence and indecency shown on television and Internet. Standards for labeling would allow consumers to make better informed decisions, to *reject* or to *retrieve* multimedia objects based on contents. *This retains our right to free expression, but gives us the means to hear and to see only what we want to know.*

Keywords: Digital products, enveloping, labeling, copyright, authentication, communication, decency, legislation.

Preface

It has taken 16 years, beginning in 1906, before the quality of food and drugs came under federal regulation with standards for their safe packaging and labeling to protect our *physical health*.

The protection of electronic commerce started with the Computer Security Act in 1987. But information technology is evolving so rapidly that we still do not know what we get when we pay for **digital products**, and find it difficult to help our children reap the benefits of audiovisual education without damage to their *mental health*.

1. Problems and Partial Solutions [2.3]

Work towards the national information infrastructure (NII) is progressing.[2] But the still uncontrolled growth of goods and services on public information highways like Internet, brings into doubt their suitability for business and profit. As television, computers, and communications merge, and compete for buyers of advertized goods, the lack of accountability and safety leads to large-scale illegal copying, fraud, a plethora of proprietary 'secure' payment schemes, and concerns about the unbridled access by minors to adult material. Industrial espionage and the threat of information wars suggest the need for a fundamental solution.[3] The narrative below is meant to point to the root of the problem.

1.1 Lack of Accountability on Public Information Highways

Some welcome the unregulated growth of information technology and credit it with the spectacular rise of a new industry. The entertainment industry stretches its limits, and computer products get sold without liability for their end-use. Software is still not 'recognized' as a commercial products by the Uniform Commercial Code (UCC). Warranties promise *only* replacement when found to be defective, or not conforming with documentation upon receipt. All license agreements substantially include the following language:

"... To the maximum extent permitted by applicable law, originators or suppliers of products disclaim all other warranties, either expressed or implied, including but not limited to implied warranties of merchantability and fitness of the software and its documentation for a particular purpose ...

... originators and suppliers are not liable for any damages whatsoever (including without limitation, indirect damages, consequential damages, and damages for loss of business profits, business interruption, loss of business information, or other pecuniary loss) arising out of the use or inability to use this product, even if originators or its dealers have been advised of the possibility of such damages."

Even security products to protect assets are being sold this way. Costs for product maintenance and loss of business are difficult to assess. Some estimate that the diminishing security of financial transactions by traditional means alone costs taxpayers more than \$10 billion each year; in part due to skilful counterfeiting of financial instruments, the fraudulent replication of bank checks with magnetic toners on laser printers, net scams, and the compromise of credit cards or cellular phones. For 1995, the GAO reports some 250,000 unauthorized penetrations of unclassified Pentagon networks.

In the absence of federal guidance for a safe and interoperable electronic payment system, ingenious proprietary schemes promise safe traffic and payments over unsecured lines. [4] Developers of the future joint multi-function smart-cards for VISA, Master Card, and EuroPay plan to keep their method secret.[5] Stored-value cash cards already competed successfully during the Summer Olympics in Atlanta. The prospect of anonymous transactions is a concern for vendors of copyrighted material to unknown buyers who may pose as fronts for illegal copying.[6] The Treasury Department questions the potential loss of taxes due to unregulated electronic commerce and *anonymous* trans-border flow of funds. The Department's issue paper on cyberspace taxation is not expected before late this summer.

In Europe, smart-cards are already used to carry personal insurance and medical information. But in the United States, the application of authentication tokens with memory is driven by incompatible domestic products and an invasion of foreign technology. For military service, the much needed replacement of the 'dog tag' with a smart Personal Information Carrier (PIC), containing not only the name, rank, and serial number, but also training records, MOUs, security clearances, medical emergency data, etc., is stymied in the absence of a trusted cryptographic chip.[7]

1.2 Infringement and Piracy of Intellectual Property [8]

Intellectual property has always been difficult to protect. Under the law, copyright protection is afforded all producers and assignees for original art. But the transition of their representations from paper, to analog and digital notations has made possible indistinguishable replication.

Printed books and literary works are registered with the Library of Congress with the issue of a unique ISBN number that is dated and quoted on each title page. International copyright agreements serve the industrialized nations, but do not deter large-scale illegal copying of books in developing countries.

Analog audiovisual products now bring more royalties to the entertainment industry with protective marks on their video tracks to prevent illegal copying on replay equipment equipped with the 'Sierra Chip.' Predictably, clever minds found ways to circumvent this filter and numerous decoders get advertised by mail order for their use in the privacy of homes. Reports by the British Copyright Council to the Commission of European Communities show alarming statistics of piracy. [9]

Digital products still get mass-produced as identical copies on cassettes and CDs for music, movies, interactive games, entire operating systems, and software like *Windows-95*. Microsoft Corporation has tried to reduce losses by holographic emblems on the *outside* of each package sold. But these external visual marks appeal to the integrity of buyers and do not interdict their illegal use. For each genuine copy sold, at least one copy is said to be made illegally. Recent surveys reported at last year's NCSC conference, imply that 40% of software in U.S. businesses is pirated, and nearly 90% of all software used elsewhere.[10]

These losses of revenues from electronic publishing can be avoided, as was demonstrated already in 1990 by the U.S. Veterans Administration when it delivered the 2.3 million medical records of veterans to its field offices, encrypted and compressed, on one CD. The now patented technology packages the data together with its controlling application and boot records on identical disks. Distributors can unlock for the consumer any of the data elements and processing commands when authorized or paid. The technology utilizes entity-to-entity authentication with multiple keys over dial-up lines or Internet. [11] This type of electronic publishing eliminates the tedium of transferring large files by wire. ORACLE Corporation is using as similar approach with success.

Derivative technology can retain proof of ownership by sealing into the medium, at the time of sale, the transaction parameters: Seller-ID, Buyer-ID, time-stamp, etc. But, in the absence of an enveloping and product description standard, the CDs from each vendor must be used with their distinct proprietary application program.

1.3 Dissimilar Approaches for the Registry of Intellectual Property

To keep up with the number and complexity of digital objects, agencies responsible for the registry of intellectual property are switching to electronic filing and collaborate with contractors, industry and academia to develop disparate tools for object identification in support of their mission:

- Copyright Office Electronic Registration, Recordation & Deposit System (CORDS)
- Patent and Trademark Office (DOC/PTO)

CORDS is a recent initiative to automate the Copyright Office Electronic Registration, Recordation & Deposit System of the Library of Congress.[12] Started in 1993, it is to automate the registration of literary works and future hypermedia objects. Despite the practical applications of the standard general markup language, SGML, and its subset for hypertext, a new, more powerful meta language is envisioned with the help of academia. The future CORDS language is to be capable of locating compound objects stored on a distributed network by reference to its registered unique 'handle' which defines its compound properties. No presumptions need to be made about the object or its location. The CORDS approach intentionally avoids issues of content. This is intended to make it flexible and extendible for registration and should permit the assembly and retrieval of very large and complex mosaics of registered hypermedia works for research and education. As to content, users will be expected to scan the retrieved objects with data analysis tools, or use additional meta constructs, similar to those for surfing on the World Wide Web. The Corporation for National Research Initiatives (CNRI) and the Interactive Media Association (IMA) provide support.[13] Applicants will use *Privacy Enhanced Mail (PEM)*. www.cnri.reston.va.usa & www.ima.org

The Patent and Trademark Office (PRO) plans to speed up the processing time of patents from more than two years to one year, in order to cope with a 6% increase per year and 236,679 patents in 1995. After experimenting with a *proprietary* system developed in Visual Basic by the European and Japanese patent offices, the PTO is now converging on SGML. This will facilitate the effective use of Internet, because 40% of all patents in the United States are filed by foreign companies, and because 35% of patents are filed by 50 large companies and firms that already use SGML and the preferred *standard* markup language for the description of intellectual property.[14]

In the absence of an agreement how best to describe hypermedia objects, agencies spend their limited resources on their immediate needs, deferring to deal with the complexity of a unifying standard at later time.

1.4 'Indecent' Programming and Multimedia Technology [15]

Audiovisual programming with interactive feedback is an effective tool for instruction. First developed for military training and concurrent engineering, visualization in three dimensions leads to faster understanding and better designs. In schools, academia and industry, video instruction with feedback is complementing personal interaction with teachers and experts. Individual, computer-aided instruction is revolutionizing the historic education in classes.

The entertainment industry commercialized virtual reality in video games with spectacular effects. Some defend that vicarious experience relieves stress and is beneficial. But following the bombing in Oklahoma City, the FBI found detailed illustrated instructions how to assemble and trigger the type of explosive used. It had been freely circulated on Internet for some time. This prompted strong concern about the availability of anti-social and subversive literature, violent depiction of crime, and adult audiovisual material on Internet, cable, and television.

Minors are known to spend hours watching TV and playing interactive computer games each day. Parents find it difficult to stop their children from learning time and time again how to hurt, maim, rape, kill, and use drugs or alcohol to resolve conflict. Confronted with similar situations in real life, they react by reflex. This seems to be confirmed by the recent *"The UCLA Television Monitoring Report."* [16] To date, all imitation crimes are said to have been settled out of court. Self-regulation by the Motion Picture Association of American (MPAA) has helped [17], but civic-minded observers point to a slide towards greater gore and progressively base and promiscuous programming by producers to retain their clientele. This is dispassionately documented by the movie critic Michael Medved as a war on traditional values in *"Hollywood vs. America"*. [18]

For an objective assessment, the government contracted for an impartial, comprehensive study documenting the increase in violent crime during the last decade, and found: Arrests of juveniles are disproportionately greater for murder, rape, robbery and aggravated assault than ever before.[19] Attorney General Janet Reno responded by stating:

"What you see here is a road map to the next generation of crime. Unless we act now to stop young people from choosing a life of violence and crime, the beginning of the 21st century could bring levels of violent crime to our community that far exceeds what we have experienced thus far."[20]

Congress included in its recent overhaul of the Communications Act the V-Chip and enacted provisions to restrict 'indecent' exposure of minors to graphic depiction of violence, crime, and porn on television, cable and Internet. But courts upheld free speech as a guaranteed right,[21,22] further confounding the confrontation between Administrative policy and the needs of industry and consumers to benefit from electronic trade. [Table-3]

1.5 World Wide Concerns

Leading educators and law enforcement officials attribute the growth of frivolous crime worldwide to violent TV programming, extravagant interactive shoot-and-kill games, and comics which hold the attention of children and teach them how to inflict harm and kill with a laugh: *"Just point and click - Bang!"*

In the United States, sexually explicit movies can not be shown on television and Cable. US Code 18P 1462 and subsequent legislation prohibits interstate transportation of obscene material, with lesser restrictions intrastate. Violent and spectacular videos are consequently offered for sale or rent in local stores and get sold for broadcast overseas. Our country is the leading information society, but the entertainment industry is perceived as an inadvertent promoter of crime:

- Canadian regulators debated last year the adoption of a screening system against TV-Violence that would include 'blacking out and filtering' offensive U.S. Cable channels in Canada during prime time.[23]
- President Mitterrand is quoted as not minding so much his young French generation to be infatuated with English-speaking videos, but being abhorred by the pollution of their minds, and the learning of reflex behavior that may lead to violent crime, torture, sex and drug addiction.[24]
- German Prosecutors investigated whether CompuServe was violating pornography laws in Germany by making 'indecent' resources on Internet available through its online service.
- The Peoples Republic of China announced earlier this year the intent to filter 'perverse,' violent, and pornographic material from Internet at national interconnecting gateways.
- Islamic countries are attracted by western technology, but they are appalled by our uninhibited culture and its freedom of violent expression. They worry about the ill effects of uncontrollable MTV and rock music with anarchistic lyrics and seductive imagery upon their children.
- Russia and former adversaries are embracing democracy as a new and better way of life, but are critical of our permissiveness and corollary export of violent crime, allegedly as an unavoidable penalty of a people blessed with the constitutional guarantee *"of free expression to say and to show as they please."*
- General Agreements on Tariffs and Trade (GATT) came to a halt in Geneva during the recent deliberations when countries could not agree on uncontrolled export of U.S. broadcasts and videos. The issue was tabled.

Trying to explain these accusations, observers point to recent adds by foreign firms that advertise still greater doom and gore in their latest video games. They question whether losers of WW-II may have bought controlling interests in the U.S. entertainment industry to corrupt our youth, and thus to defeat us at home, if not in battle. If true - it is unlikely to work as planned: *The genie left the bottle and is taking global flight.*

1.6 Threat of Industrial Espionage and Information Wars [25]

Most commercial software is still being sold as a collection of dated files, named in a packing-list. Buyers have little assurance about their completeness and integrity. Once installed, the application depends upon other components, like the operating system, and perhaps firewalls. Even when object-oriented design constructs are applied, the attention

of developers is likely to focus on interfacing and operability - not on integrity and security. This practice for software carries over into the design and fabrication of integrated circuits, components, and functional assemblies.

The United States is the world's largest consumer of electronic equipment and audiovisual products, of which 70% get manufactured in the Far East.[26] U.S. business and U.S. weapons depend on them. But dormant, known faults and embedded hostile algorithms in software, hardware, and integrated circuits can be triggered by remote controls to cause havoc and denial of service.[27] It will take time before critical ICs get equipped with unique identities to permit assured links to other components with certified entity-to-entity authentication that is no longer *bit*-sensitive.[28] For mission-critical applications, a new class of trusted, resilient systems will have to be invented. [29]

CIA Director, John M. Deutch, expressed these concerns in his recent public testimony to a Senate subcommittee, in June, when he urged Government and industry to prepare for cyber-warfare attacks on U.S. computers. Noting that military and civilian systems are highly vulnerable to such attacks, the Senate Governmental Permanent Subcommittee on Investigation was told "*we have evidence that several countries are developing the doctrine, strategies and tools to conduct information attacks. These warfare techniques could disrupt such critical services as utilities, air traffic, and finance, with very large onslaughts likely within a decade.*" [30]

The recommendation of *The President's Commission on National Infrastructure Protection* is still awaiting approval. According to recent reports, President Clinton is not expected to issue the executive order, for a multi-agency commission, to plan a defense against cyberwars by foreign governments and terrorists, before 1997.

2. Emerging Standards point to the Feasibility of a General Solution

The *Computer Security Act* of 1987 was meant to correct some of these problems. In 1989, following the tests of public-key algorithms by DoD/OSD's PLUS program for the Protection of Logistics Unclassified Systems, we expected public-key standards and products to follow and serve a myriad needs. In 1990, we asked the *Computer Security Systems and Privacy Advisory Board* to hasten the process with a 'comprehensive mapping' of *technical* and *legal* issues, e.g., those characteristic of *handwritten* signatures and *digital* signatures, to learn what must be done, and what can be done better. But it took four years of debate for the DSA to get confirmed, and a draft of legal issues for "*Digital Signature Guidelines*" can be ordered this year from the American Bar Association (ABA).[31]

Standards are a basic requirement of electronic commerce. All committees are at work to combine data authentication and encryption into an enabling technology system for simple and complex tasks. It is an arduous undertaking. The national and economic welfare of each country is at stake - and evil knows no bounds. New and simple solutions *will* rise, but for the problems of today, we have to use what we have. For the purpose of this report, we note the accomplishments, and recent recommendations of CALS, NIST, and NSA committees to ISO and UN/EDIFACT for enveloping, labeling, and the *Security Service API for Associated Data*.

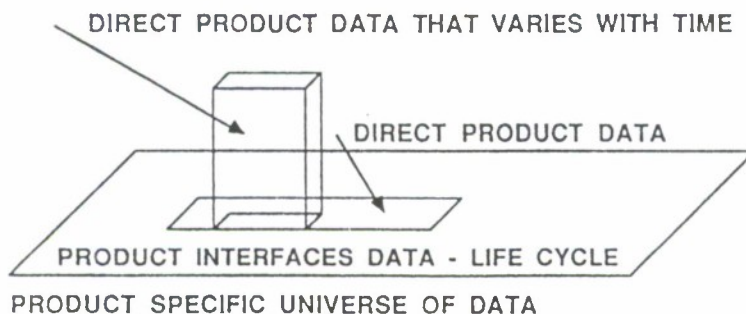
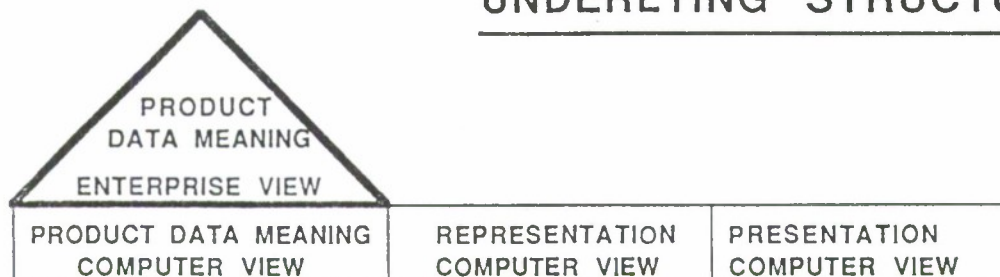
2.1 CALS Standards for Enveloping, Description, and Authentication

CALS is a strategic approach for standardization, known worldwide as the standard for Continuous Acquisition and Life-cycle Support. Started by OSD in 1984 at LLNL in collaboration with 120 contractors to prepare for the delivery of modern weapon systems.[32] Concurrent engineering and automated manufacture are the new goal. This pooling of talent makes product integrity and communications security a necessity, and can no longer be pursued in isolation. In 1994, the Secretary of Defense, directed the Department to save costs by releasing *all* U.S. *military* specifications to industry for the procurement of less expensive, *commercial* products with state-of-the-art technology. CALS military specifications now serve a global industry.[33] (Table-1)

Enveloping, and universal classification of digital objects is the key to Phase-2 of CALS. The requirements stem from the early understanding that an integrated weapon systems data base would have to carry in digital form an object through its entire life cycle of design, analysis, manufacturing processing & planning, the ordering of spares, and

online provisioning. These requirements are complemented by the need for technical and training manuals, maintenance instructions, automated manufacture, logistics, and support analysis. Objects get created only once, but are available to hundreds of collaborators from government, contractors, and military components - with selective and time-dependent authorizations for access and use. Presentations of objects are unique collections of product data, described by their preferred constructs [34]:

UNDERLYING STRUCTURE



PRESENTATION (IE DOCUMENTS, DESIGNS) ARE UNIQUE COLLECTIONS OF PRODUCT DATA - AND SHOULD NOT BE STANDARDIZED

Description of data is critical. The CALS enterprise requires classification of objects, universal data dictionaries, translators, and interpreters for control procedures, capable of recognizing and removing inaccurate, untimely, and inadequate data, and equipped to add, revise, explain, and authenticate correct data. If one is unable to categorize data, one is unable to establish their complete unambiguous meaning. This requirement led to a family of generic data description languages, embodied in the Mil-Std-1840 enveloping standard.

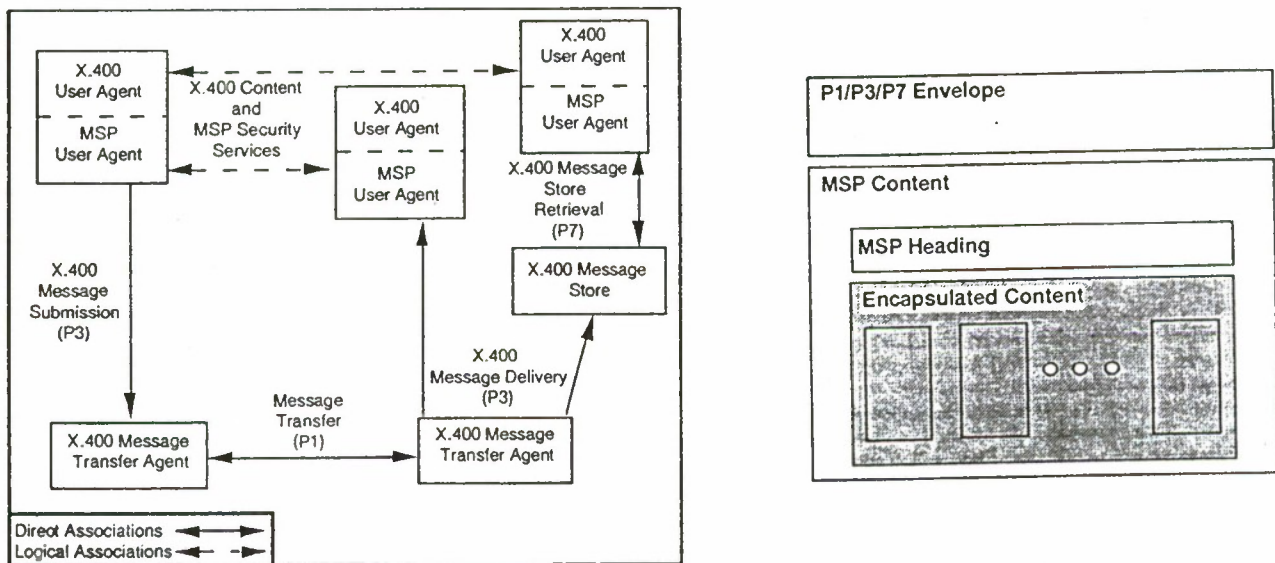
The results are impressive. The *PDES* product definitions exchange specifications are the cornerstone of CALS. *SGML*, the powerful CALS standard general markup language, led to the popular subset for hypertext markup, *HTML*, and *JAVA*, which permit cross-platform portability. Earlier this year, *JAVA* has been selected to give near-universal portability for the next generation of *WordPerfect*. The fusion of text, graphics, video, and audio now extends in CALS to the standards for photography and motion pictures, *JPEG* & *MPEG*.

Authentication has been added to Mil-Std-1840. It now includes digital signatures and ANSI X.509 key certifications. Version-3 was submitted by the DISA Standards Office to ISO for review.[35] Results of the modernization and standardization enterprise are being tested continuously on the CALS Test Network (CTN), headquartered at LLNL, and get reported at international CALS conferences and in committee work. The CALS enterprise and its approach to data manipulation has universal appeal, and enjoys global endorsement. In the United States, the OSD CALS office, NSIA, and CALS Resource Centers transfer CALS technology to industry and vice versa; CSC and SRA corporations work jointly with other contractors and industry on security and integration of the CALS enterprise for DoD.[36,37]

2.2 Message Security Protocols (MSP) for the Transport of Digital Objects [38]

The early availability of federal and commercial E-Mail may explain in part why the need for a generic, universal packaging and description standard for digital objects did not arise earlier. Most E-mail has provisions for the attachment or encapsulation of digital objects with varied levels of security. Some have cross-platform and cross-algorithmic compatibility, and interface to *Fortezza*. But most do not provide inherent message security, and are *not interoperable*, unless they comply with the CCITT x.400.

The need for writer-to-writer secure and interoperable E-mail service over public networks, and the requirement to treat each encapsulated object with distinct levels of attention, let NSA to develop the *Message Security Protocol (MSP)* as the recommended standard for interfacing with the international *CCITT X.400 Message Handling System*. MSP is an integral part of the Secure Data Network System (SDNS) and is documented in SDN.701 (V4). Recent revisions in SDN.701 (V4) permit any type of message to be sent and received securely, including the ANSI defined X.400 Message Transfer System conventions for electronic data interchange (EDI). Of particular interest for this report is MSP's capability for the secure encapsulation of *any* digital object as shown below:



Encapsulation of Digital Objects in the MSP Protocol

Until a universal, secure object description standard is developed, *secure encapsulation* is a viable alternative. We quote from SDN.701: The MSP user agent resides between the X.400 user agent, and the X.400 message transfer agent, and X.400 user agents may be either distinctly or tightly coupled protocol entities. MSP is independent of the message content being protected, and is independent of the user's message preparation system. The new content type, MSP, is submitted to the X.400 message transfer system. For message delivery, the recipient user agent may either form a direct association with the message security agent or may use a message store. This message store provides the information on delivered messages to support selective processing, including: connectionless confidentiality, authentication, integrity, access, control, and non-repudiation with proof of origin and delivery. The NIST Standard Security Label for Information Transfer, FIPS PUB 188, can be used to convey this information from the encapsulated object to MSP and X.400. The SDN.701 (V4) recommendations have been forwarded to ISO for review.[38,39]

2.3 Security Service API and Recommendation of their Extension to Digital Objects [40]

NSA's Cryptographic Application Program Interface (CAPI) recommendations go one step further. They express the urgent need for a general, unifying approach not only to the Message Security Protocol, but also to the prevailing cryptographic modules of commercial secure-mail products, and their integration with applications. Developed by NSA's Cross Organization CAPI team since last year for *Application Programs*, NSA's recommendations should be generalized to *Digital Objects*. A justification would read nearly verbatim like that given by NSA for CAPIs:

"Until recently, the integration of cryptographic functionality into digital Object Descriptions has required that developers tightly couple the *Object* to the cryptographic module. This approach forces each new combination of *Object* and cryptography to be treated as a distinct development effort, and does not provide for the modularity and maintainability expected of commercial products. An approach that can provide flexibility and cost savings is the use of a standardized *Cryptographic Object Description Interface (CODI)* suite.

The compelling case for a modular cryptographic interface has given rise to the development of proprietary CODI proposals. An NSA cross-organizational team should be formed to assess the ability of these proposals to meet anticipated needs. A review of the CAPI criteria should be made as to their applicability to *Digital Objects* in general, leading to a detailed analysis and recommendation, etc...

... Rather than recommending a single *CODI*, the NSA team may find it appropriate to recommend several that could handle simple and complex classes of *compound objects* with needs for distinct cryptographic service of their subsets, similar to the proposal for the CAPI standards, GSS-API (Internet Engineering Task Force), the GCS-API (X-Open), CryptoAPI (Microsoft) and PKCS #1 Cryptoki (RSA), because these CAPIs had been designed to support significantly different levels of cryptographic awareness, ranging from minimal security needs to extensive requirements for the underlying cryptography. Criteria to be considered by the team to assess each of the predominant CODIs should also include: algorithm independence, application independence, cryptographic module independence, modular design and auxiliary services, MISSI cryptographic support, safe programming, and security perimeter designs.

Even though a *CODI suite* may have to be recommended to address the needs of a wide variety of digital objects, it is anticipated that most digital objects and products will require minimal knowledge of the underlying cryptography. Therefore, the NSA team should be encouraged to select the (one) high-level CODI that best serves present and future needs."

2.4 ISO and UN/EDIFACT extensions to "Associated Objects"

In the meantime this summer, UN/EDIFACT sub-committees have been authorized to update their syntax for "associated" data in Part-8 of the standard. It is to provide a framework for more general, unspecified 'associated data sets' capable of accepting any bound digital object and its notation, for which some allowance is made already in ISO 9735. The updates will include encapsulation of the EDI standard transaction sets and the ANSI x.12.58 data security structures. (www.R3.ch/sjwg)

Personal communications with committee members suggests that updates of standards *react* to market pressures. A *proactive* approach is needed for the proposed system of standards for the encapsulation of digital objects, their structures, content, need(s) for protection, and binding. Security and data labels could serve to transfer requirements for cryptographic service, and to the object transport service, similar to the approach taken by NSA for the recommended standardization of the *Security Service APIs*. [41,42]

Good electronic mail systems replicate and exceed the services of postal mail. Digital signatures can do more than handwritten marks. But in retrospect, these capabilities have come about by trial and error. For electronic commerce, a deliberate proactive approach should map societal conventions for the delivery of goods and services to those expected from electronic commerce. A Congressional initiative would encourage collaboration to bring it about.

Summary Statement

Legislation for electronic commerce of digital products and their communications is inevitable. *Ddigital objects* have to be bound and protected by *digital* means.

Standard digital labels in a structured header would protect producers, and give consumers the means to select, use, and pay only for products they need at work, approve for their families, or desire to see as adults.

A *Protection Act for Digital Objects* would encourage collaboration. It should cover all aspects of electronic commerce and extend to communications, television, cable, and public networks. Consumers should be able to ascertain by recourse to the x.509 public-key directory that an acquired object was genuine as advertised.

When President Bush was first briefed on this proposal in 1989, it was too early. Today, it is technically and economically feasible.[Table-3]

Supreme Court Justice Clarence Thomas, when he heard of this approach during an informal meeting, stated emphatically:

"This technology could give us for the first time the right to hear and to see - only what we want to know!"

U.S. DoD	INDUSTRY	PURPOSE of STANDARD and its APPLICATIONS
MIL-HDBK-59		Provides guidance on the technology, standards, and procurement process as related to the transition from a paper intensive activity to one operating with digital information
MIL-STD-1840		This is the primary standardization document for the selected CALS standards. It identified, by application, which industry standard and which corresponding DoD standardization documentation to use. It also provides standard "enveloping" procedures for transferring standard data forms.
MIL-D-2800	IGES	The Initial Graphics Exchange Specifications (IGES). It represents a neutral file format for the representation and transfer of product definition data among CAD/CAM systems and application programs.
MIL-M-2801	SGML	A Standard Generalized Markup Language (SGML). It contains markup requirements, tagging, and generic style specifications for page-oriented document text.
MIL-R-2802	CCITT Group 4	Specifies the efficient compression of scanned raster images. It uses the code from the Group 4 facsimile recommendation of the International Telegraph and Telephone Consultative Committee (CCITT). A "tiled" form is described by using the architecture nomenclature of International Standard, IS 8613.
MIL-D-2803	CGM	Computer Graphics Metafile (CGM). A neutral format for the description, storage, and communication of graphical information.
EC/EDI	EC/EDI	Electronic Commerce/Electronic Data Interchange (EC/EDI). The electronic interchange of business information between trading partners. Users standard formats currently defined by ANSI X12 in the U.S., EDIFACT in Europe, and AECMA 2000 for NATO.
STEP	STEP	Standard for the Exchange of Product Model Data (STEP). A computer-interpretable data representation format being developed to include all product model data necessary to define geometry, function and behavior of a product during its life cycle. Product Data Exchange using STEP (PDES) in the U.S. standards activity supporting STEP.
MIL-STD-974	CITIS	Contractor Integrated Technical Information Service (CITIS). Specifies the contractor-provided service for electronic access and/or delivery of contractually committed business and technical information with need-to-know.
MIL-M-87268	IETM	Prescribes the requirements governing the creation of Interactive Electronic Technical Manual (IETM) and the development of IETM presentation software, applicable to a computer-controlled Electronic Display System (EDS).
MIL-D-87269	IETM	Prescribes the interchange format for the delivery of an IETM database to the Government.
MIL-Q-87270	IETM	Prescribes the requirements for an IETM Contractor's Quality Assurance (QA) program.
MIL-HDBK-SGML (draft)	SGML	Provides guidance in the application of MIL-M-28001, which is based on ISO 8879, Standard Generalized Markup Language. Data prepared in accordance with these guidelines will facilitate the automated storage, retrieval, interchange, and processing of technical documents from varied data sources.

Table-1: DoD CALS Standards evolve into International CALS Standards

Table-2: Precedents for the Labeling of Consumer Products

All legislative powers to establish justice, insure domestic tranquility, provide for the common defense, promote the general welfare, and secure the blessings of liberty to ourselves and our posterity is vested in the United States Congress. The list below highlights some legislation, its implementation, and industry's response to public concerns about what we *eat, hear, and see*.

- 1906: The Food and Drug Act (21 USC 1-15), and its implementations by the Food and Drug Administration (FDA), now requires content labeling for foods, drugs, and cosmetics.
- 1922: Movie Production Codes are distributed with films made by the Motion Picture Association of America (MPAA) in a voluntary program to describe their content for the benefit of the viewing audience.
- 1968: Movie Ratings G, M, R, and X are introduced by the MPAA to replace the rudimentary 'production codes' with better and simpler definitions, as self-regulation in response to criticism by the public that some codes were unfit for children. R prohibits attendance of children *below* 16, and X of children *below* the age of 17.
- 1984: The M-Rating for Mature audiences gets split into two rating, PG and PG-13, requiring stronger Parental Guidance for the PG-13, which recommend no admittance under the age of 13 years.
- 1990: The X-Rating gets upgraded by one year into NC-17: No Children 17 and under admitted. This is in response to the U.S. Supreme Court decision because of greater violence and crime in X-movies. States and cities receive some local controlling power. MPAA ratings get printed *externally* on video containers (of different colors), and are shown at the start, not during, movies containing indecent material unfit for children.
- 1992: Cable Television Consumer Protection and Competition Act of 1992, restricts indecency on Cable.
- 1994: The White House announces *Clipper* and *Skipjack* to protect voice and data, including key-escrow.
- 1996: The V-Chip to filter Violence gets added to the revised Public Communications Bill by request of President Clinton. MPAA representatives promise implementation in TVs and desktops within a year.

The Communication Decency Act (CDA), restricting indecency on Internet, was ruled in June by a three-judge federal court in Philadelphia to be unconstitutional, because *technology* of Internet did not allow people who post information, to control who may receives it. The Justice Department is appealing.

The Cable Television Consumer Protection and Competition Act of 1992, restricting indecent programming on Cable, was ruled in June by the Supreme Court as unconstitutional in part. Cable providers may refuse indecent programming (*sexually explicit or patently offensive*) on leased channels paid for by independent programmers, but may *not* refuse to air indecent programming on public access channels, when requested by local governments and educational organizations. (e.g.: Information on AIDS, Abortion, etc.)

FM-side band labeling and control gets started as a voluntary technology to label different radio broadcasts, and their associated advertisements tailored to different 'classes' of listeners with perceived needs and means.

The Platform for Internet Content Selection (PICS) is a control product by MIT's WWW-Consortium (AO, CS, Prodigy, etc.) for providers to label resources so that parents can exclude them with PICS products. www.w3.org/pub/WWW/PICS/. Some sites require credit card numbers or 'adult' PINs.

A draft standard for Key-escrow of stronger exportable encryption is promised by NIST/CSL to help resolve the debate between federal agencies and industry.

As television and telephony migrate to *digital* broadcasting, *digital* controls should become technically and economically feasible, utilizing technology developed for the communication and control of *digital* data.

Table-3: Origins and Chronology of the Proposal

"To fail is hard. But it is worse, never to have tried, to succeed" - Theodore Roosevelt

- 1984: William Taft IV, Deputy Secretary of Defense, launches the *CALS* initiative by issuing a memo requiring defense industries to deliver the definitions of 'how to build and maintain' new weapon systems in digital form.
- 1987: The Technology Information Systems Program at LLNL, under contract to OSD and WPAFB, completes the first draft specifications for the enveloping, and labeling of digital products in **MIL-Spec-1840-A** and turns it over with the first draft of SGML to the new CALS Standards office at NIST, headed by *David Betwy*.
- 1989: NIST improves the specifications and releases them as the **MIL-STD-1840-B** standard.
- 1989: President Bush receives a 10-minute briefing on the proposal as a means to regulate violence on television by extension of the military CALS standards for enveloping and labeling to the neutral marking of audiovisuals. The "dual-use" proposal is given by Chase Undermeyer, at the request of the presidential election team, during a marathon session of alternative solutions for *Critical Issues*, Jan. 7th, 1989.
- 1992: President Bush and Vice President Quale get briefed again on the proposal by the presidential reelection committee. Supreme Court Justice Clarence Thomas, during an informal meeting on the constitutionality of the proposal: *"This retains our right to free expression, but gives us also the means to hear and see only what we wish to know!"* Sen. John Danforth receives the proposal, forwarded by Clarence Thomas.
- 1993: Digital Signature and Application Symposium, NIST. "Dual-use Initiatives: "Digital-Labeling of Digital-Products." Dr. Peter Weiss, Policy Analyst, Information Branch of the Office of Information and Regulatory Affairs, remarks that the proposal would be more balanced if extended to the labeling of intellectual property rights.
- Information Highways for the National Information Infrastructure (NII), Privacy Aspects, Public Hearing, Washington, DC, Dec. 27, 1993. The proposal is enlarged for consideration as a *Digital Products Protection Act*, patterned after the *Food Protection Act.*, Formal submittal to Patricia Faley, DOC Director of Consumer Affairs.
- 1994: William J. Perry, Secretary of Defense, directs the release of military specifications *"to facilitate the development of dual-use processes and products for the benefit of an expanded industrial base that is capable of meeting defense needs at lower cost."* DoD's *"Computer Aided Logistics System"* becomes the global *"Continuing Acquisition Life-cycle Support"* Vice President, Albert Gore Jr. receives the proposal in January, via Mike Nelson.
- 1995: Public meetings and submission to NIST, NSA, OMB, and the Committee on Commerce, Science, and Transportation:
- Public Workshop on Key Escrow & Data Authentication, NIST, Gaithersburg, MD, September 6-7, 1995.
 - Development of Federal Key Escrow Standards, NIST, Gaithersburg Hilton, MD, September 15, 1995.
 - Photonics-East: Voice, Data, and Communications, Symposium on Data Protection, Philadelphia, Oct. 22, 1995.
 - PKI Infrastructure of the NII, SI-PMO, Public Meeting, GSA Auditorium, Washington, DC, Nov. 3, 1995.
 - 64-bit Software Key-Escrow Encryption Export Criteria, NIST, Gaithersburg, MD, December 5, 1995.
 - Senator Larry Pressler, Chair, Hearings of the Communications Bill, November 9, 1995.
 - Senator J. James Exon, Minority Leader, Hearings of the Communications Bill, November 12, 1995.
 - Committee on Commerce, Science, and Transportation, Patric G. Link, Chief of Staff, Hearings on 'On-line Pornography' and the Communications Bill," November 9, 1995.
- 1996: All candidates for President.

References

1. President William C. Clinton, Vice President Albert Gore, Jr., *February 22, 1993. Technology for America's Economic Growth, A New Direction to Build Economic Strength*, GPO, ISBN 0-16-041692
2. *National Information Infrastructure, Security Initiatives*, Proceedings, NISS Conference, 1995.
3. National Academy Press, National Research Council, Washington, DC, (800) 624-6242:
The Unpredictable Certainty, Information Infrastructure Through 2000, NII Steering Committee, 1996.
Keeping U.S. Computer and Communications Industry Competitive, Comp-Sc. & Telecommunications, 1995.
Realizing the Information Future, The Internet and Beyond, Comp-Sc. & Telecommunications, 1994.
Rights and Responsibilities of Participants in Networked Communities, Comp-Sc. & Telecommunications, 1994.
Computers at Risk, Comp-Science & Telecommunications Board, 1994
4. *NetCash*, Practical Electronic Currency on the Internet, Medvinsky & Newman, USC/ISI, ACM Infosec Proc.93.
5. *Smart Card Forum '95*, Proc., Tysons Corner, VA, September 18, 1995, Hq.: Tampa, FL, (813) 286-2339.
6. *DigiCash*, Prepaid Smart Card Currency Techniques, D. Chaum, Amsterdam, Netherlands, +31-20-665-2611
7. *MARC, DoD's Multi Application Resource Card*: 3-of-9 bar code, magnetic stripe, embossed data, printed information, digitized photograph, and IC with 2 KBytes of memory. POC: Mike Noll, Pentagon, 703.614.1996.
8. *Intellectual Property at a Crossroads, Global Piracy and International Competitiveness*, Congressional Leadership Institute, Stephanie Epstein & James M. Jones, 1990.
9. Melissa J. Shaw, *Software Piracy: Prevention, Detection and Liability Avoidance*, Proc. 1995 NISS Conference.
10. Software Publishers Ass.(SPA), *\$8.8 Billion Loss Due to Software Piracy*, Newsbytes News Network, 02-19-95
11. G. Lang, The Harrison Corporation, *US Patents 5,065,429 (1991) and 5,191,611 (1993)*. (301) 622-3177.
12. Copyright Office, Library of Congress, Washington, DC 20559-6000, (202) 707-8350, Ms. Mary Levering.
13. Corporation for National Research Initiatives, Reston, VA (703) 620-8990, Dr. Robert Kahn and Bill Arms.
14. Government Computer News, *On-line Patent Filing is Critical*, June 15, 1996.
15. *The First Amendment and the Fifth Estate*, Regulations of Electronic Mass Media, T.B. Carter, Foundation Press
16. *UCLA Report on the Effect of Television on Teenage Behavior*, November, 1995.
17. Motion Picture Association of America (MPAA), 1600 Eye Street, NW, Washington, DC 20006, (202) 293-1966
18. Michael Medved, *"Hollywood vs. America"* Harper Perennial, 10 East 53rd Street, New York, NY 10022
19. Melissa Sickmund et als, *"Urban Delinquency ..."* September, 1995. Call 1-800.638.8736, Report # 153569
20. Washington Post, *Arrests Soar for Violent Crime of Juveniles*, A14, September 8, 1995.
21. Washington Post, *Court Upholds Free Speech on Internet, Blocks Decency Law*, A1, June 13, 1996.
22. Washington Post, *Supreme Court Frees Cable-TV for Explicit Programs*, A10, June 29, 1996.
23. Washington Post Editorial Column, *TV North of the Border*, September 18, 1995.
24. National Geographic Society, July 1989 issue.
25. Win Schwartau, *Terminal Compromise*, Inter.Pact Press, *Information Warfare*, Thunder's Mouth Press, 1994.
26. Technology and Assembly Research Council (TMRC/AMRC), June 16-17, 1994.
27. V.Hampel, *Corruption & Detection of ICs with Hostile Algorithms (U)*, Keynote HOTEL Conf. Pentagon 05-20-90
28. V.E. Hampel, *Protected Interoperability of Telecommunications and Digital Products*, Critical Reviews, Defining Global Information: Infrastructure, Systems, and Services, Vol.CR56, SPIE Press, (360) 676-3290.
29. Miscellaneous publications on DoD's C4I Program
30. Government Computer News, *Deutch: U.S. Systems will be under fire in 10 years*, July 15, 1996.
31. ABA, *Digital Signature Draft*, M. Baum, Esq., 750 N. Lake Shore Dr., Chicago, IL, 60611, (312) 988-5599.
32. LLNL, echnology Information Systems Program (TISP), V.Hampel, D.Grubbs, and C.Hatfield, (510) 422-8567
33. William J. Perry, *Specifications and Standards - A New Way of Doing Business*, June 29, 1994.
34. R. Schuldt, Lockheed Martin Marietta, Denver, CO.
35. *DISA Standards Office*, JIEO/JEBE, 10701 Parkridge Blvd., Reston, VA 22091, (707) 735-3568, Alan Peltzman Mil-Std-1840-V3, *CALS Enveloping & Data Description Standard* 06-30-96; DISA, A.Peltzman, (703) 735-3568
36. DoD/OSD, *CALS Office*, Skyline-2, Suite 1600, Falls Church, VA 22041, (202) 681-7626, Elaine F.Litman
37. National Security Industrial Association, U.S. CALS Industrial Steering Grp, Washington, DC, (202) 775-1440.
38. *Message Security Protocol (SDNS/MSP)*, SDN.701 (R4), R.P. Fanara, NSA, Ft. Mead, MD 20755-6000
39. *Standard Security Label for Information Transfer*, FIPS PUB 188, 1994, NIST/CSL, Gaithersburg, MD 20899
40. *Security Service API: Cryptographic API Recommendation*, E-2, 07-07-96, NSA Cross Organization CAPI Team
41. *The Computer Law and Security Report*, May/June 1995, Vol. 11, Issue 3, ElSevier Advanced Technology.
42. *"Global Law for Commercial Digital Products,"* J. Ritter, Chair ABA/EC, OSCE/ECLIPS, (614) 292-6082.

THE BUSINESS-LED ACCREDITOR - OR....

HOW TO TAKE RISKS AND SURVIVE

Michael E J Stubbings
Room A/1411,
Government Communications Headquarters,
Priors Road,
CHELTENHAM
Gloucestershire
GL52 5AJ
United Kingdom
Tel: +44-1242-221491 ext 3273

What The Accreditor Does - But Shouldn't

The computer security accreditors inspect a new computer system. They have previously gone through System Security Policies (or Plans) with a fine-tooth comb. They have ensured that every 'i' is dotted and every 't' crossed. They have put their feet down with firm hands all the way through the project - with semantic contortions to match. The system manager is on his best behaviour - with all the more troublesome users sent on leave for the day. Sample audit trails are available containing evidence of carefully staged 'security-related' events. The accreditors prow around the system, looking stern, as is expected of them. And then a certificate is signed - the system is now accredited: it meets the rules. Everyone is happy.

What have we achieved? Time is money, and we have spent a lot of it in giving this system its certificate. We may well have bought hardware or software products solely to satisfy the rules. We are likely to have imposed ways of working on the user community which they would otherwise not have implemented. We now know that this system, its operating and configuration control procedures, all meet the rules. Which is what the accreditors have traditionally been for - to ensure that systems meet the rules.

This approach has a number of advantages. These include:

- a) Clarity: everyone knows where they stand. Systems either meet the rules or they don't.
- b) Documents: In the USA the Orange Book, and in the UK, the CESG (Communications and Electronics Security Group) Memoranda give all the guidance necessary.
- c) Training: Low training costs, as rules are easier to teach than judgement.
- d) Culture: This approach fits well with traditionally rule-based or hierarchical environments.

Life being what it is, there are some disadvantages. These include:

- a) Support: A large infrastructure of developers, evaluators and accreditors is needed to support this approach.
- b) Perception (1): Security is perceived to be a hurdle - no sense of local 'ownership'.
- c) Perception (2): No perception of accreditation as an instrument for obtaining business advantage, i.e. value for money.
- d) Costs and Benefits: Rules and procedures do not reflect the value of the assets (systems or data) to the organization, nor the costs of the different sorts of security breach.
- e) Value: No definition of 'value'.

To this I would add a few personal observations. I was for many years a system and project manager - on the receiving end of the accreditors' ministrations. In November 1993 I became the senior computer security accreditor, at just about the same time that a new head of computer security was appointed - my immediate boss. Apart from the above, we both noticed that:

- a) Accreditors (expensive people) spend most of their time at their desks reviewing documents.
- b) Whenever an accreditor spoke to a system or project manager, it was usually to tell them that they had done something wrong.
- c) There was distrust, suspicion, and occasionally open (and verbally robust) hostility between accreditors and system/project staff.
- d) In an increasingly value-driven environment, the concept of justifying imposed security costs did not exist. Some of the measures we imposed did not add anything to a system's security profile. They were imposed because the rule book (or custom and practice) said they must be imposed.
- e) Neither end-user objectives nor system functionality (that is, the system's value to the organization - its business case) had any place in the accreditation process.
- f) Accreditors were overworked to the point that individuals were suffering, and there was an increasing danger that systems with real security problems were being 'lost in the noise'. Conversely, most systems and projects presented very few real problems (as opposed to theoretical ones).
- g) The commercial environment was talking about risk management; about quantifying and assessing risk. We didn't normally use the word 'risk'.

Of course, a lot of us *do* use that word, but how many of us find out the risk to a system by looking it up in a table? How many of us go on to minimise that risk by looking up a series

of measures in another table? And how does that help us to know the actual vulnerabilities of our systems rather than the theoretical ones? How does that help us to assure our organizations that we are causing money to be spent wisely? That is the starting point for the GCHQ (Government Communications Headquarters) approach. Although we are in the public sector, we no longer believe that we can go to our financial planners, or to our project fund holders and say 'Spend x thousand pounds or dollars, or x project hours because we say so - trust us, we're professionals'. That isn't good enough, and rightly so. It's not an approach I would like to try getting past a shareholders' meeting or a public accounts committee.

The New UK Government Security Philosophy

At about the same time that my boss and I moved into the accreditation world, the United Kingdom Cabinet Office (similar in some ways to the various Presidential offices) issued the Review of Protective Security (RPS). This document, formally announced in Parliament by the Prime Minister, mandated a new approach throughout government service. It covered a wide range of security considerations, setting out a philosophy which changed the whole basis upon which security professionals approached their jobs. The subjects included personnel vetting, paper controls, and a range of other matters, including IT Security. It comes down to one thing. In the past we did our best to avoid risks. Now we manage them.

The background to this approach is basically what I have already been describing. Her Majesty's Government (HMG) demands value for money from its officials. Civil Servants should not spend - or allow to be spent - money which does not add something to the value of the product. Value is defined as the extent to which the product furthers the business objectives of the organization. Is security one of the organization's business objectives? In the case of my own department, the answer is most definitely 'Yes'. In other departments, particularly those holding information about individual people (e.g. Social Services, Agriculture or other ministries), the answer will also be 'Yes'. Once accepted as a business objective, security becomes the responsibility of the organization, and everyone in it, not simply the preserve of people seen often as 'those professional obstacle-makers and blame-distributors in the security department'. Sometimes we are even viewed as the people employed to *take* the blame for security problems.

The other result of this approach is that security funding has to have a business case made for it, in competition with all the other requirements for spending. This is as it should be. Perhaps a particular security measure is essential to the survival of the organization. Perhaps the cost of not implementing that measure is outweighed by the benefits of using the funds elsewhere. *Security spending is primarily a management matter, not a technical one.* If the organization gains no significant benefit from a security measure, why spend time and money on it? And if you are spending time, then you are also spending money. Those of you who work for commercial organizations will be very familiar with this approach. It has not, until now, been part of the government culture in the UK. I suspect that this vocabulary will not be entirely unfamiliar to those in US government service.

So, the idea had come of age. Government policy and our own internal observations coincided in both timing and content, and we had a marvellous opportunity to rethink our whole approach to IT security and to accreditation. We were not the first in the field (if you will excuse the pun) - the United Kingdom's Ministry of Agriculture, Fisheries and Food (MAFF) preceded us with an added-value philosophy - not that I knew it at the time. We didn't stop with IT. What I am about to describe was carried on within a total rethink of the functions and tasks of an

internal security division. IT Security does not exist in a vacuum; it shares an environment with paper-handling, personnel, training, and procedural security measures. If there isn't a common philosophy for all of these, with an obvious relationship to the organization's culture and shared objectives, security measures become discredited, circumvented and imposed only by force. Under those circumstances, no-one wins.

What We Did

We didn't call it Business Process Re-engineering, but that's basically what we were doing. We took the RPS philosophy, and looked for the core processes which would further GCHQ's business objectives, and defined what contribution those processes would make. We then set about designing a structure and set of procedures which would implement these processes with the greatest economy and efficiency - in other words achieving the maximum value for money. I'm not going to describe the way we went about doing this, save to mention that we involved our client community - GCHQ's project, system and security managers. Many interviews were carried out, and it was interesting to note that the observations noted earlier were largely consonant with what our clients were saying. The one quotation which sticks in my mind is that the computer security branch staff were 'A bunch of computer illiterates with a six-inch rulebook'. We are not that, and never were, but it shows the extent to which people on both sides of the accreditation/project divide had stopped listening to each other - if they had ever started. The fact that our clients had said that about us showed that regardless of the RPS, something had gone seriously wrong.

What We Ended Up With

At the end of all this soul-searching, we came up with a set of principles, an environment for them, and tools with which to apply them. Part of the environment was a 'given' - the physical nature of the GCHQ campus, the physical and logical aspects of the department's existing telecommunications, the law of the land, and the policies of HMG. Most of the rest was open to us to reshape as we saw fit - and we did.

The Principles

Our principles are unlikely to come as a surprise to anyone; they came directly from the RPS philosophy and from our own observations.

- a) IT security is the direct and accountable responsibility of the system users and managers, it being by definition part of their overall security profile and therefore one of their own business objectives - an idea often abbreviated to the concept of 'local ownership'.
- b) The accreditor's job is to assist project and system staff to identify, document and accommodate their own security risks and requirements, where by definition these include GCHQ's corporate requirements, and then to certify if they have been met.
- c) The actual provision of IT security features and procedures is *not* the accreditor's job.
- d) Each security measure must add value to the system, where value means that the

cost of the measure is exceeded by the consequent business benefits. Accreditors must therefore identify security-related proposals which are not cost-effective, with a view to their removal.

e) Security costs include impediments to convenient use, limitations to desired functionality, security and system administration overheads, and the costs of extra hardware, software or maintenance contracts.

f) It is essential that IT security staff are available as advisors to system managers and their users throughout the life of the system.

g) It is essential that the organization has some assurance that despite the move away from rule-based accreditation, appropriate and cost-effective corporate standards are identified, adhered to, and kept under periodic review.

The Environment

I have already alluded to the 'given' nature of part of GCHQ's environment. A particular set of site access rules, security patrols, personnel clearance policies etc. were already in place. For obvious reasons I am not going to describe these: suffice to note that the existence of a well-established and reliable campus-wide regime allowed us more flexibility in the construction of our procedures than might otherwise have been the case. I would add that the TEMPEST profile and risk assessment associated with the two GCHQ sites in Cheltenham is an important factor in defining the environment within which we operate.

Aspects of the environment which were open to adjustment and renewal included our own structures, staffing, job descriptions and internal IT resources. When we went into this process, we had one senior computer security accreditor (me) with five assistants. Two of my staff concentrated largely on collaborative projects, i.e. those where GCHQ's internal policies did not apply because of the involvement of other agencies such as the Armed Forces. There were, in addition, 2 Computer and Communications Security policy staff who for historical reasons undertook various infrastructure and communications accreditation tasks. When considering our structure, we also had to bear in mind the wider security division reorganization which I mentioned earlier. As it happened, the two programmes dovetailed nicely, and the new structure reflects the requirements of both.

Our New Structure

We redeployed one accreditor to lead a Computer and Communications Security Monitoring Team, and recruited two assistants for her. They have a two-fold job. The first is to carry out a security inspection of each area in the department, such that everyone can expect an inspection every 2-3 years. The scope of each visit is all staff, systems and procedures operating under a particular security management regime. That usually means one open-plan office, or a contiguous group of offices or laboratories. The visits are intended to be advisory in manner, so that they can work with staff to enhance their security effectiveness, rather than coming in as a police force trying to catch people out. Naturally, disciplinary procedures are available to deal with wilful disregard of security measures, but we are not interested in pursuing people for honest mistakes and misunderstandings. We would rather sit down with them and help them to improve matters - for the sake of their own, and therefore corporate, effectiveness. The Monitoring

Team's job is to ensure that systems continue to be configured and operated in a manner reflecting their declared and approved security profiles.

The second role for the Monitoring Team is as an incident response office. Should a suspicious IT security event be noted, it will be investigated first by local staff, who are obliged to call in the Team if a satisfactory explanation is not immediately forthcoming. Team members have a wide variety of resources to call upon to support them. These include the accreditors, the department's own technical experts, staff from other security disciplines, and members of the Communications and Electronics Security Group (CESG). CESG is the UK national authority for communications and computer security matters, setting guidelines for all government systems. They are collocated with, but separate from, GCHQ. It is broadly similar to the USA's NSA/ISSO organization. At the time of writing, we are considering the possibility of seeking liaison membership of FIRST (Forum of Incident Response and Security Teams) for our Monitoring Team.

The Monitoring Team coordinates closely with the Internal Audit Unit. Each acts as a specialist adviser for the other, and they take care that their respective inspection programmes do not clash. Reports from each are made available to the other, insofar as personnel and management data release considerations permit this. In practice, such factors should rarely apply.

All other computer and communications accreditation work (including that previously carried out by the policy staff) is now handled by the remaining four accreditors, plus myself. What might have been an unmanageable increase in workload is assuaged by a change in procedures limiting the amount of attention given to routine systems. This procedural change is described in more detail below. I am using this opportunity to redefine my own work pattern in order to devote time to more general topics such as defining a security profile for a GCHQ Corporate Web (that is, one most definitely *not* connected to The Internet). The two policy staff are moving into a dedicated policy unit serving the interests of all the security disciplines. As and when IT Security policy issues arise they will coordinate task-orientated teams drawing on, among others, staff from the accreditation and monitoring teams.

The two teams are located in adjacent offices - with an open door between them. They share IT resources, including system databases, and a common office automation environment. Both teams report to the same senior manager. It is our intention that a close working relationship should continue between the two groups.

The Tools

Our experience and measurements led us to believe that something like 75% of the incoming accreditation workload related to systems which either presented no real security threat, or were operating in arenas where appropriate security profiles had already been defined. It therefore made little sense for accreditors to handle each system individually. In order to reflect this, and to implement the principles defined earlier, we decided to put all systems into one of two categories: routine and exceptional.

Routine Systems

These systems are the 75% just mentioned. They operate within a clearly defined security profile. This profile includes the system's location, the classification (or protective marking as

we say in the UK) of its software and data, the clearance level of its users and managers, and its connections. A flowchart was drawn up to guide system and project managers to a decision as to whether or not their systems fell within this profile. For those which do, a campus-wide document set was written, comprising Baseline Security Measures, department-wide Security Operating Procedures (aimed at system and security managers), and a department-wide Secure Features User's Guide (aimed at the normal user). These are all very short documents, setting out the security objectives in functional terms, plus the responsibilities of individual members of staff. These include responsibilities for configuration and change control, system management procedures, and also define the circumstances under which reaccreditation would be required. Project and system managers wishing to have a system accredited are asked to confirm in writing that they accept and can implement the measures described in these documents. If so, they register their systems with the accreditors. The system is then entered into the Monitoring Team's visits programme, and an accreditation certificate is issued. For the first six months of this new way of accrediting systems (starting January 1996), the Monitoring Team will in fact inspect every routine system, in order to verify whether or not the new methods are working effectively. As I said earlier, systems needing attention were in danger of being lost 'below the noise'. The introduction of a 'routine system' accreditation track will reduce the noise level to a point where we can handle the systems which would most benefit from our attention.

Exceptional Systems

That leaves the systems which are 'interesting'. These continue to be handled in the classical manner, for the most part with a tailored document set, considerable accreditor involvement at all stages of the project, and a detailed post-installation inspection. It is, of course, open to the accreditor to use any of the routine system document elements should they be deemed suitable. Some systems will be exceptional for reasons connected more closely with administrative considerations rather than security ones. I anticipate an increasing level of formality when holding commercial data - you may remember the presentation last year entitled 'The Development of Generally-Accepted System Security Principles', which addressed this issue among others. Other systems will present problems, where some new balance of procedural, technical and personnel controls has to be found in order to achieve a satisfactory security profile. Perhaps money has to be spent, perhaps the functionality has to be redefined, perhaps the accommodation needs to be altered. Maybe it's a simple matter of adjusting the system configuration. In all of these considerations, the accreditor has to find the appropriate cost/benefit balance. Once this balance is found, an exceptional system will usually be inspected by the accreditor, possibly in the company of a member of the Monitoring Team. It will then be entered into their continuing inspection programme.

Summary

Rule-based, predominantly technical, computer and communication security measures are no longer a cost-effective response to the security requirements of a modern organization, whether in government, commerce or in industry. If an organization has inadequate security, its business effectiveness is impaired and its survival threatened. If an organization has too much security, it is wasting resources. That too will limit its business effectiveness and threaten its survival. Security must make its case for a slice of the corporate cake along with all the other business activities, and it must make its case on the basis of its contribution to the overall well-being of the organization. Security is first, last and always a management matter, whether the management is at the level of a national government, or the board of directors of a small company.

Technical measures exist only to implement business objectives effectively, at minimum cost. This is what GCHQ has sought to implement using the mechanisms outlined in this paper. At the time of writing (early February), we have just implemented the change, and we think we have got it just about right. By the time of the 1996 Conference, we will know for certain. I suppose I'm therefore taking a risk by submitting this paper in advance of a settling-in period. Still, risk management is what it's all about.

INTEGRATION OF DIGITAL SIGNATURES INTO THE EUROPEAN BUSINESS REGISTER

Helmut Kurth

Industrieanlagen Betriebsgesellschaft mbH
Einsteinstr. 20
D-85521 Ottobrunn, Germany

kurth@iabg.de

Abstract:

In the INFOSEC programme 1994 the European Union set up three trial projects to demonstrate the feasibility of the use of digital signatures in pan-European networks. One of those trials was the EBRIDGE project, where customers could extract the official business register data from companies of four European countries on-line and authenticated by digital signatures. Authenticated data from those official business registers has to be presented in some European countries to notaries or banks for specific types of contracts. Today the common way to obtain this data in an authenticated form is to get an officially signed copy of the business register data by surface mail. Since it may take up to two weeks to obtain this information in this conventional way, some contracts were delayed by this time and financial losses could be the result of this delay. With the infrastructure established in the EBRIDGE project, the official business register data can be obtained digitally signed in a few seconds. In addition, the EBRIDGE projects demonstrated that the official business registers could also serve as Trusted Third Parties by maintaining public keys of company representatives and distribute them in a secure way.

Introduction

In 1994 the European Union set up the INFOSEC'94 programme which had the main objective to demonstrate the use of digital signatures and trusted third party services in pan-European trade. Three projects were started covering different aspects that are needed to establish a Public Key Infrastructure in Europe. One of those projects was the EBRIDGE project, which integrated digital signature technology into the prototype of the European Business Register. The European Business Register tries to link the official business register data bases of the countries within the European Union and makes this data available by on-line services.

In all European countries, each company has to register itself to an official authority before it can start to operate. Entering data into and changing data in those business registers is performed by authorized persons only (e. g. notaries). Data in this register contains among others: the name and address of the company, the legal status of the company, the names of persons allowed to sign contracts on behalf of the company and additional information about the business areas and some financial information about the company. These official business registers play an important role for trade in Europe. Many countries demand that for specific types of contracts the signed copy of the official business register data for all companies involved in the contract has to be obtained and attached to the contract. Banks often also request such a signed copy of the business register data for

credits. Obtaining manually signed copies of the business register information is time consuming, can slow down business and may even result in lost business opportunities.

In 1993 four European countries (Denmark, France, Italy and the UK) started a project to make their official business register data available on-line and link those databases together. This project was called the European Business Register (EBR) project and was funded by the European Union under the ENS programme. It was soon recognized that on-line access to business register information without a proof of authenticity of this information was not sufficient. Therefore in 1994 the EBRIDGE project was started to enhance the EBR prototype by integrating digital signatures. The partners within this project were: Mercury (UK), Cerved (Italy), OR-Telematique (France) and DCCA (Denmark) as the Service Broker in the four countries that run the EBR trial, Sema Group (UK) as project coordinator, Denton Hall (UK) and ISTEV (Italy) dealing with the legal aspects, IABG (Germany) leading the security architecture design and implementation and Siemens Austria who provided the digital signature software.

Overall architecture

The European Business Register (EBR) provides a European Union-wide public service for the retrieval of officially registered information concerning European companies. It has established a running prototype network that currently includes four European countries (France, UK, Italy and Denmark). In each country there is a service broker, who provides the on-line access to the local business register data. EBR interlinked those national service broker thereby allowing customers to access business register data from all four countries.

Figure 1 provides an overview over the current structure of the EBR.

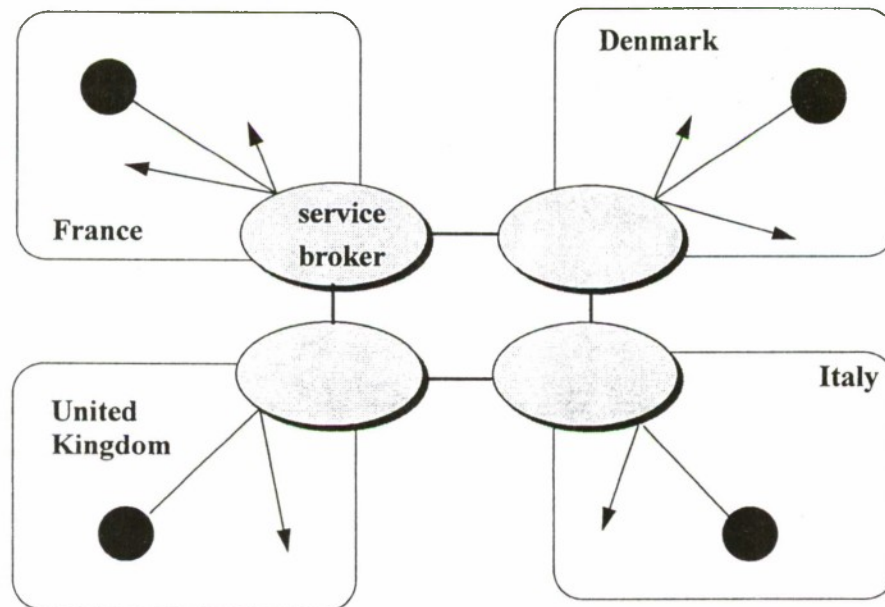


Figure 1: EBR Service Infrastructure

The figure shows that each customer contacts only his national service broker, which extracts information from other countries via the EBR network and transforms it to the presentation style specific its own customers. Each service broker has a direct link to a national on-line registry database system. These database systems are the sources of all retrieved information. The service

broker are interlinked using an X.25 network. A proprietary protocol was designed on top of X.25 for the information exchange between service brokers.

The goal of the EBRIDGE project was, to provide the customers of EBR with the possibility to prove the authenticity of the business register data. Several problems had to be solved within the project:

- The structure and content of the business register data bases in the four countries is different due to the different laws concerning the business registers in each country.
- Since the electronic business registers had been developed separately, different query languages are used in the different countries.
- Query results from other countries have to be transformed into the national presentation style. Since different languages are used in the four countries, field names and the content of some fields has to be translated before it is presented to a customer in another country.

This requires that the query as well as the data resulting from the query has to be transformed on its transmission path. Since this transformation would invalidate the digital signature applied by the originating service broker, the data has to be re-signed by the local service broker before the data is transferred to the customer. To maintain the chain of trust from the customer to the originating service broker, the local service broker maintains an audit trail of all signed data he received from a foreign service broker. The data he re-signs and transmits to his customer contain a 'link field', which points to the audit record of the original data he has received from the foreign service broker. In case of a dispute, he can immediately extract the signed original data from the audit trail and put the responsibility for the correctness of the information on the originating service broker. This general architecture is shown in figure 2.

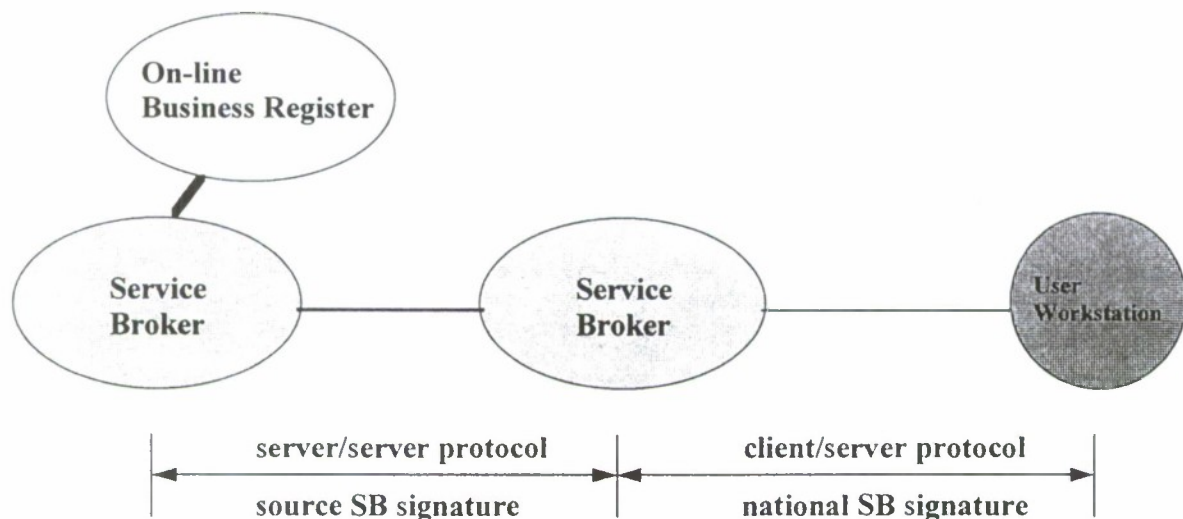


Figure 2: Service Broker Signature Functions

Each service broker domain consists of several systems serving different purposes:

- A service broker host (usually a large mainframe system) which contains the business register data base

- A Unix based machine, called the 'Portable Service Broker System' (PSBS), which interconnects the different service broker systems via a X.25 based network
- A special 'Protected Electronic Signature Machine' (PESM) operating in a protected environment which generates and verifies digital signatures.

The Portable Service Broker Machine was specifically developed to allow other nations to join the EBR easily. This machine provides a simple adaptable interface to the service broker host to allow the integration of different types of host and their database systems to be integrated into the EBR network. The system was developed on a Unix basis and can be ported easily to different Unix based platforms.

The Protected Electronic Signature Machine was also developed on a Unix basis. The Unix operating system was specifically configured and extended to allow secure operation. The specific features of the Protected Electronic Signature Machine are described in more detail later in this paper.

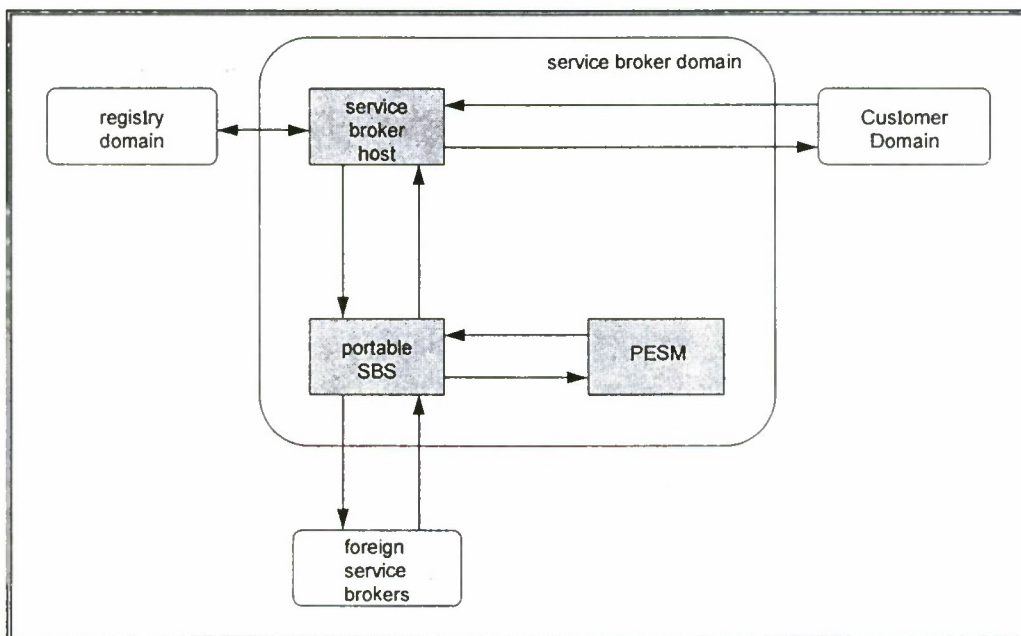


Figure 3: Structure Of A Service Broker System

A simplified form of the flow of information resulting from a user's request is shown in figure 4. The sequence of information flow is shown in the diagram by the numbered arrows.

The user is the originator of all requests for service. A service request may specify the delivery of signed or unsigned information. The request itself is not signed. Requests for delivery of signed information are handled in the following way. Information is signed before transmission by the SBS providing the information and the signature is verified upon receipt by the SBS requesting the information on behalf of the user. This second SBS then transforms and re-signs the information and forwards it to the user workstation. This extends the chain of trust backwards, giving both the user and the local SBS confidence that the company information has not been modified since it was signed after retrieval from the distant business register – and did indeed come from the appropriate

service broker. A user's trust depends upon the local SBS having an appropriate level of security to protect the international transfers and so the service brokers have to be trusted.

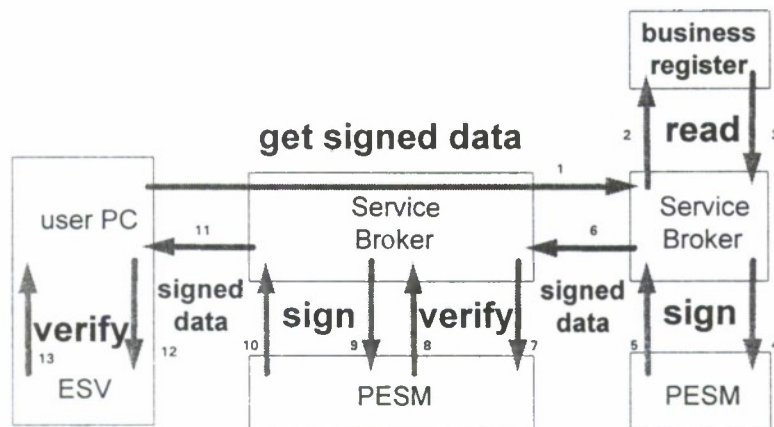


Figure 4: Flow of Information in the EBRIDGE Architecture

To explain the semantics of the arrows in figure 4, we take the example of a UK customer, who wants to obtain authenticated business register data of an Italian company.

1. The user submits the query request to the UK service broker. The UK service broker transmits the request (after transformation to the Italian request format) to the Italian service broker.
2. The Italian service broker submits the request to the Italian business register database.
3. The reply from the Italian business register database is sent back to the Italian service broker.
4. The Italian service broker sends the data to his Protected Electronic Signature Machine to digitally sign the data.
5. The data and the signature are passed back to the Italian service broker system.
6. The signed data is then transmitted to the UK service broker.
7. The UK service broker submits the signed data to his Protected Electronic Signature Machine to verify the signature of the Italian service broker.
8. The PESM tells the service broker system, if the signature was successfully verified.
9. When the signature was verified successfully, the service broker system writes the signed data to an audit trail and transforms the data received according to his national presentation scheme. Then the transformed data is sent to the PESM to be signed by the UK service broker.
10. The PESM of the UK service broker signs the data and passes data and signature back to the UK service broker system.
11. The UK service broker system passes the data to the customer, who has submitted the request.

12. The customer locally verifies the signature of the UK service broker.

13. After successful verification of the signature, data and signature are stored in the customer's system.

For the EBRIDGE service signatures are created using a one-way hash function (MD5) and asymmetric encryption (RSA). Key pairs are generated within the PESMs at each site. Private keys are isolated within these PESMs. Each service broker has to pass its public keys to each other service broker and each service broker has to distribute its own public key to its local national terminal population. A specific protocol has been implemented to allow the service broker to change their key pair. The certificate passed with each signed information contains the date of issue of the key used to sign a document. Whenever a customer or a foreign service broker detects that he does not have the current public key of a service broker, he may issue a request to submit the new public key. The service broker then submits the new public key signed with his 'Master Key'. Since this master key is used only to distribute the new public key, a longer key length can be used for this key (e. g. 2048 bit). This and the fact that only few messages are signed with this key reduces the possibility of a cryptographic attack against this key.

The other crucial element in the EBRIDGE architecture is a trusted audit trail. Each service broker maintains an audit log containing records of received signed information. Information forwarded to a customer contains a reference to a specific record in the audit log of the local SBS. In the event of a problem or dispute these audit logs can be used to determine the point in the delivery chain at which the problem arose, so identifying responsibility for problem resolution.

The Protected Electronic Signature Machine

As a result of the risk analysis undertaken in the beginning of the project it was decided to have a separated Protected Electronic Signature Machine instead of maintaining the private keys in a general purpose machine to which many people may have access. Separating the electronic signature functions from all other parts of the application allows for a high level of physical and procedural protection which minimizes the risk that the private keys are compromised or that the critical software is tampered with. The design of the PESM allows to concentrate all the critical hardware and software components of a service broker in a single machine which performs only the security critical actions of the service broker. Only a limited number of persons must have physical access to the machine.

The operating system of the PESM is Unix. It was chosen because it can be configured in a way which restricts each user of the PESM to only those functions he needs for his operational role. The PESM supports only three operational roles:

- System Administrator
- Key Administrator
- Auditor

The Unix system is configured in way which prohibits that any user can log in as root or obtain root privileges. Only users with one of the special application defined roles mentioned above are allowed to log into the system. If additional maintenance actions have to be performed, the operating system has to be reconfigured thereby deleting the critical data stored in the system.

Backup procedures for this data exist, which automatically encrypt the backup medium. Two persons with different roles (System Administrator and Key Administrator) are needed for backup and restore where each person defines a part of the key used for encryption. The encrypted keys on the backup medium are of course integrity protected.

Each user is assigned one of the roles mentioned above. For each role a restricted shell has been developed which will allow each role to execute only those commands necessary for their task. None of these roles will directly get superuser privileges. All actions which require special privileges (e. g. adding new users or changing a user account) will be performed by setuid-protected programs. Especially all actions of the key administrator will be performed in a way which prohibits that the key administrator has direct access to the private keys stored in the machine. This has the advantage that although the Key Administrator is able to generate a new key pair on the PESM, he is not able to read the private key he has generated.

An audit function is installed which logs each attempted login as well as each command issued by a System Administrator or a Key Administrator. This audit log is maintained by the Auditor and can not be accessed by either a System Administrator or a Key Administrator.

These functions and features ensure that the PESM can be operated securely. But of course these functions need to be complemented by physical protection, procedural measures and measures to assure that the security critical software in the PESM is implemented correctly and has no security critical side effects. The physical and procedural protection features are described in a System Security Plan that has been produced as part of the project. This plan states all the measures that are assumed to be sufficient for the provision of a very high level of security as required by a commercial service. In addition a pre-evaluation using the ITSEC and ITSEM was performed for the critical application software of the PESM.

Conclusion

The EBRIDGE project demonstrated the use of digital signatures for the on-line access of data from official registers in a field trial, that was conducted in the summer of 1995. The functions can be seen as a prototype for the on-line access to other types of registries containing data that has to be authenticated. In addition the EBRIDGE project can be a starting point for a Public Key Infrastructure for commercial business in Europe. If the public keys of persons authorized to sign contracts on behalf of a company are stored as part of the data in the official business registers the EBRIDGE system would provide the service brokers with the ability to act as a Certification Authority for those public keys. No new infrastructure is needed for this purpose.

Technically the EBRIDGE project has solved the problems to establish a part of a Public Key Infrastructure that can be used to for Electronic Trade in Europe. But currently the legal situation is not yet prepared for such a system. Digital signatures are not yet accepted as equivalent to hand written signatures and the legal status of the official business registers is not identical in each European country. But with the demonstration of the feasibility of integrating digital signatures and Trusted Third Party services in an existing infrastructure the EBRIDGE project pushed the demand for a solution to the legal problems.

The EBRIDGE project is just one example of several projects the EU has funded to establish a base for electronic trade in Europe. Other projects in this area are the projects FAST, BOLERO, TESTFIT and CAFE. Within its new ACTS programme the EU has started two other projects

(SEMPER and FANS) that will help to establish a working public key infrastructure that can be used for electronic trade in Europe.

Also the legal situation is beginning to change. In several countries new laws are under discussion which set up the legal framework for the acceptance of digital signatures. E. g. in Germany a proposal for such a law has been distributed to technical and legal experts for discussion. This proposed law contains also several technical requirements regarding the security of the systems operated by a Trusted Third Party as well as the security of the user equipment that generates the digital signature. The technical security requirements for systems operated by TTPs fit well with the features implemented in the PESM within the EBRIDGE project.

References

- [1] INFOSEC'94 Workplan, Commission of the European Union, July 1993
- [2] S2201 EBR ES&TTP Phase 1 Final Report, EBRIDGE Project, May 1994
- [3] FAST II Project Description, Philips, April 1994
- [4] EBRIDGE ES&TTP Enhancement to EBR, Final Report, EBRIDGE Project, August 1995

Industrial Espionage Today and Information Wars of Tomorrow.

October 22-25, 1996
Baltimore, Maryland

Paul M. Joyal
President
INTEGER Security Inc.
Information and Analytic Services
725 15th Street, NW, Suite 908
Washington, DC 20005
Tel: (202) 347-2624, Fax (202) 347-4631

ABSTRACT

In this report we review case histories of industrial espionage publicized in the media and in Congressional hearings. The threat to the United States as the world's largest investor in R&D is magnified by the transition from a cold war military confrontation of the super powers to economic competition in global markets. To sustain their market share, France, Japan and Russia have initiated national programs to acquire U.S. technical know-how. Former intelligence staff now distill fragments of sensitive information into meaningful intelligence to guide industrial and national efforts towards dominance. This threat is amplified by the exponential proliferation of global communication networks, like the INTERNET, that reach into corporate America and permit unseen adversaries to probe the vast U.S. data stores for unprotected intelligence. Counter intelligence in industrial espionage by the United States on a national level is virtually impossible because of public scrutiny in our open society.

On the positive side, the upheaval of a rapid transition from the global political high-tension and high stability of the Cold War to the low-tension and high instability of the so-called new world order has prompted increased international collaboration against international terrorism and organized crime. On the corporate level, strategic alliances with foreign firms are expanding in order to sustain competitiveness and innovation in areas of specialty. A national security plan to protect the U.S. information resources is needed; and a viable policy to enable our information highways to operate as safe conduits for electronic business. The well being of our nation or that of the global economy, should not be left to chance and provocation.

KEYWORDS: Industrial espionage, economic espionage, case histories, terrorist attack, disaster prevention, corporate alliances, national collaboration, information warfare.

1. INTRODUCTION

National Competition in Global Markets is well illustrated by the following example: In June, 1990, US intelligence officials learned that Indonesia was about to award a contract for roughly \$100 million to the Japanese electronics firm NEC to modernize its creaky telephone system. Authorities familiar with the specifics say that AT&T's subsidiary in Europe had a more competitive bid and that Tokyo used its \$2.1 billion in foreign aid to Jakarta to sway the decision in NEC's favor. It took the intervention of President Bush in a letter to President Suharto, to have the bid split between the two firms. As David Boren, then Chairman of the Senate Intelligence Committee remarked, "The world has changed ... We have to change with the world."¹

Moreover, one of the reasons that McDonnell Douglas has considered entering into a foreign ownership arrangement with the Taiwanese is that it is becoming increasingly more difficult for McDonnell Douglas to compete against the likes of the French *Airbus* passenger plane. The rise of *Airbus's* share of the world market to 30% has largely come at the expense of McDonnell Douglas. Some company officials claim that if they can't get an industrial policy in the U.S. they must go to where they can get one. McDonnell Douglas recently lost a sale to Singapore Airlines when Airbus sweetened the offer with scarce landings rights at the two airports in Paris.² These are examples of how economic wars are presently being waged.

In his 1987 book **Real-World Intelligence: Organized Information for Executives**, Herbert E. Meyer contends that throughout the world of commerce and industry, "intelligence" is on its way to becoming a key management tool for the corporate executive.³ Meyer goes so far as to suggest that the emergence of *business intelligence systems* is the most striking and potentially significant business trend of our time. Here, intelligence becomes the means by which companies chart their future course and are alerted to the strides of their competitors. As economic competition intensifies, and as it redirects the activities of covert agents who were previously focused on military competition, the importance of acquiring information concerning economic plans and intentions will become more acute. Correspondingly, systems and methods for the protection of proprietary information will also be elevated in priority and importance.

Meyer reminds us that successful business enterprises have been collecting and using intelligence for centuries. Examples of the organized use of intelligence can be found as far back as the fourteenth century in the historical accounts of the *House of Fugger*.⁴ One of the first European international banks, the *House of Fugger* produced a "manuscript newsletter" for its key officers, what today might be described as a "competitive intelligence" newsletter. This newsletter contained succinct political insight and sensitive commercial information. What differentiates these efforts of the past with those of the present is the organized nature of the correlation of the material. In the past many elements of business intelligence have traditionally been carried out in only the largest and most successful of companies. Today the effort is to integrate that information and to disseminate its analysis and conclusions to those decision makers who are expected to use it as a managerial tool. "It is this new effort within the corporate community to acquire, organize, and coordinate the diverse elements of intelligence that is turning a group of related, but previously separate activities, into a new, and incredibly powerful business management tool."⁵

Today the world is changing before our very eyes. Some old rivalries are fading while others are coming to the forefront. Today's friend may become tomorrow's enemy and visa versa. This is especially true in the rapidly changing world of shifting political alliances and changing boundaries for economic competition. Secretary of Defense William Perry has warned: "We live in an age that is driven by information. Technological breakthroughs....are changing the face of war and how we prepare for war."⁶ What are we to do and who are we to guard ourselves against? The French government has been extremely focused on these questions in recent years. Findings of a recent investigative report are gradually becoming revealed.

2. THE FRENCH CONNECTION

Three years ago, in France, a high-level commission was very quietly formed under the office of the Prime Minister. The Commission was headed by Henri Marté, CEO of the French military aircraft manufacturer *Aérospatiale*. The task of the Commission was to study how a selected number of foreign countries integrate economic information obtained from various governmental and non governmental sources. Among the countries studied were: Japan, Sweden, Germany, the United States, and Great Britain. As the Commission investigated the systems which various countries have developed to integrate commercial, trade, intelligence and *company-proprietary* information with *open-source* information, they found France lacking. They concluded that France, while having many of the desired components, suffered from a lack of focus and failed to integrate the various activities and sources of information towards an effective national course of action.

The Commission for Security and Economic Competitiveness had thirteen official members, including prominent businessmen from defense companies such as Mr. Lagardere of Matra, as well as Professor Montagner at the Pasteur Institute. But the vision and forceful drive of the Commission flowed from its director Mr. Reme

Pautrat. The sixty-ish former head of the French internal security service, DST, Mr. Pautrat brought all the competence and enthusiasm he was known for in his former position; DST is the French internal security service, equivalent to the FBI in the United States. Some prominent members of the distinguished French business organization CNPF have expressed their general uneasiness concerning the efforts of its own government. They, perhaps more so than members of the former socialist government, have failed to accept the notion that economic problems or competitiveness can be anything more than marginally improved through government intervention and involvement. Some French businessmen have even openly expressed their skepticism for this government effort. Their concerns are that any government effort, coordinated through the office of the Prime Minister, will be counterproductive and lead to more suspicious attitudes against French companies in global markets. What distinguishes this planned approach from earlier activities in France is the attempt not only to aid the large French national and strategic companies in their global posture, but also to reach out to the small- and medium-size French companies on a regional basis to promote the export of their goods.

One of the concrete steps undertaken so far by the French government has been the establishment of a DECA-type education program by the DST in 1993. It reaches out to educate companies on the various threats they might face in industrial espionage and economic intelligence. They are also becoming more and more concerned about disinformation and covert action. Some suspect that Coca-Cola was behind the benzene scare that negatively effected the business of Perrier, due to the public reaction and concern about the purity of its widely-favored drinking water. No doubt, the French have many case studies and experiences to draw upon; some of these concern IBM, Texas Instruments, and Corning, as will be noted later. The fact that Mr. Pautrat's old organization is highly active, should come as no surprise. However, not to be left behind is the French DGSE: the equivalent of the United States' CIA. DGSE will continue the questionable covert and 'black bag' activities in their department known as 'Service Seven', which has been targeting American business since the 1960s. This has prompted visiting American government officials to tote their briefing books, papers and documents with them when traveling in Paris. One of President Clinton's economic aides commented that he was cautioned "...that this is France we are visiting. When it comes to economic espionage, no one does it better."⁷

In another development of the Chirac government, the DST and DGSE intelligence services have established a liaison committee to work together rather than at cross purposes. This committee will attempt to coordinate information from the Economic and Commerce Departments of the French government. One old mission within the French Foreign Ministry's overseas diplomatic establishments was the **"Post Economic Expansion"** organization which attempted to help French private businessmen with investment opportunities around the world. When the Commission found that it was largely an ineffective expense for the Foreign Service, with little or no concrete, tangible achievements, the foreign service component was re-focused to feed information back into the new French national structure for information analysis. This activity, represented within the French Embassy in Washington, DC, recently forwarded to the French Finance Minister a 26-page report, dated September 11, 1995, titled **The American System to Support Private Business**. It describes all the various components of the American government which promote American business overseas, especially the Advocacy Center of the Department of Commerce (DOC). The report even contains a photograph of their "war room". It also reviews, in some detail, the September 1994 "Operation China," when Secretary Ron Brown visited China and the successful use of the CIA to dislodge the Thompson Company from a Brazilian contract on behalf of Raytheon. These initiatives have their roots in economic modeling of the United States by French entities on computers of U.S. firms under contract to the Bureau of Economic Analysis and Labor Statistics, with the latest econometric data.

The outcome of the French study about business in the United States was scheduled to be publicly acknowledged on October 31, 1995, in Essonne, France. There, the first regional national information center was unveiled to help 300 French companies to remain competitive in global markets. The ceremony was conducted by the French Minister for Economy and Finance, Mr. Arthris, by the Chairman of the new Agency, Philip Caduc, and of course by Mr. Pautrat of the SGDN. This new organization has been given a clear focus, a defined mission and will be called the **"Agency for Diffusion of Technological Information."** This new Agency will become a fusion center for sensitive information in intelligence, business, diplomatic, and ministerial communications that reaches out to French Regional Chambers of Commerce seeking to promote French economic interests. Businesses who wish to avail themselves to this valuable information will be able to subscribe to protected and secured national data banks. While presently located under the Prime Minister office, a real power play is occurring within the French government to see what ministry will ultimately run it—Finance, Economy, Trade?

The French approach goes well beyond what many have attempted to propose as a symbiosis between national and corporate efforts to compete in world markets as well as goes beyond what many believe is appropriate. Some would maintain that only classified national security information should be protected and that only military and political government-vs-government espionage should be formally addressed. The French are attempting to integrate public and private information for offensive and defensive purposes. The presence of the Pasteur Institute on this Commission should make in the United States take a second look at this problem. For example, a U.S. pharmaceutical company that has entrusted to the FDA a \$60 million dollar product of research with all of its proprietary data, could be a tempting target for foreign intelligence collection. This information is as vitally important to a private company as a Top Secret document is to the government. It is simply not true that company proprietary information is less important or valuable than government classified information. After all, the stature of a nation depends upon the well-being of its industry. In fact, much of the technological data that foreign intelligence services target and collect are neither classified nor subject to government control. The challenge of protecting emerging, innovative technology goes to the heart of one of the strategic national security issues of our country: The competitiveness of the United States in global markets. But the French approach is more holistic than has been seen in the past. It is one which must be studied, and possibly emulated, because it is much more significant in the long term as a national strategy than the uncorrelated collection of interesting technical tidbits.

This realization is especially important to countries that find it prudent to form alliances for the gathering and exchange of specific intelligence to counter a common threat, as will be noted later on. Specifically, on May 28, 1990, Jay Peterzell, writing for *TIME Magazine* in "*When Friends Become Moles*", cautions us to the troubling reality of friendly governments aiding their domestic companies by funneling to them sensitive information gathered covertly from their allies. In fact, the article asserts that the French actively utilize a special section of their intelligence service to engage in "bag operations" jobs against American and foreign businessman in Paris. *TIME* reported that Service-7 of French Intelligence (DGSE) was set up in the early 1960's to run recruitment and collection operations against foreign firms in France. *L'Express* reported that U.S. intelligence had discovered that the DGSE had recruited assets in the European branches of IBM, Texas Instruments and other U.S. companies doing business in Europe. Incidents such as this have now become part of the staple of FBI briefings on DECA. Robert Courtney, a former IBM security official stated in the *TIME* article: "*There's no question that they (The French DGSE secret service organization) have been spying on IBM's communications and have been giving the information to Bull for years.*" Increasingly, concerns are filtering throughout the contractor community that the threat of espionage is no longer limited to the traditional Soviet Bloc adversaries. Our allies or "friendly" nations must also be viewed with caution, especially when it comes to the protection of technical breakthroughs, financial insight, and corporate proprietary information. The French national initiative to collect innovative technology from competing markets prompts all countries to recognize the value of technical know-how in the emerging information age as a national resource, with the corollary need to defend it - and to acquire it for the common good. However, it follows in many ways the precedent set by Japan.

3. THE JAPANESE HAVE COME AND ARE HERE TO STAY

The Japanese efforts to penetrate U.S. industry, to gather innovative technology and sensitive business information, have made headlines in the past. Although individual corporate officers and companies have been exposed as culprits in the media, the official industrial and economic intelligence gathering is represented by the **Japan External Trade Organization**.

In Bob Woodward's controversial book *Veil*, he reports that in 1982 an investigation was opened to uncover the "leak", when it became apparent that highly classified excerpts from the U.S. National Intelligence Daily (NID) concerning the Iraq-Iran war were being sent from the Washington office of the Japanese Mitsubishi Corporation to Tokyo.⁸ This investigation eventually led to the resignation of a United States government official. Mitsubishi, along with Mitsui, Sumitomo and Nissho-Iwai, were identified by Meyer as the closest thing to a superpower conglomerate in the business world, with regards to intelligence capabilities. The Japanese trading companies or *sogo shosha* have become the standard by which others will be judged. Understanding business intelligence as organized information has been mastered by these Japanese trading companies. They have committed the resources and created the system needed to collect, analyze and distribute intelligence to key

executives. Pat Choate, in his equally controversial book **Agents of Influence**, also refers to the intelligence gathering capabilities of Japanese trading companies "as a vast overseas information collection system." Once again, according to Meyer:

*"Indeed, every branch office of every trading company operates like a vacuum cleaner, sucking in information, statistics, documents, brochures, articles from technical and current events magazines, reports delivered at industrial and scientific conferences attended by one or another Japanese executive, and even gossip picked up at dinner parties or on the golf course. Some of these trading companies' operations are substantial; Mitsubishi intelligence staff in New York takes up two entire floors of a Manhattan skyscraper."*⁹

This raw material is then transmitted back to its headquarters where its senior intelligence and analytic staff transforms it into an intelligence product. Once the information is collated, analyzed, and synthesized, the intelligence is immediately delivered to the key executives who use it via a pre-defined need-to-know system for strategic decision making. This intelligence can then be exploited for tactical purposes. Strategic planners in Japan will also directly benefit from this intelligence product.

According to a 1987 classified CIA report *Japan: Foreign Intelligence and Security Services*, Japanese intelligence service priorities were the following:

- Intelligence regarding access to foreign sources of raw materials, including oil and foodstuffs
- Detailed intelligence on technological and scientific developments in the United States and Europe
- Intelligence on political decision making in the United States and Europe, most specifically, intelligence relating to trade, monetary, and military policy in Asia and the Pacific region
- Intelligence pertaining to internal political and military developments in the Soviet Union, the People's Republic of China, and North Korea

The report concluded that 80 percent of all Japanese-government intelligence assets were directed toward the United States and Europe, concentrating on high technology developments. The CIA report also allegedly explained the critical "intelligence gathering role" played by semi-official organizations, such as the Japanese Ministry for International Trade and Industry (MITI), the Japanese External Trade Organization (JETRO), and Japanese multinational corporations such as Hitachi and Mitsubishi.¹⁰ The CIA has assessed the quality of the intelligence gathering operations of these companies to be every bit as sophisticated as the intelligence services of smaller countries, including technical penetration and collection operations. In 1985, Professor Chalmers Johnson of the University of California at San Diego estimated that JETRO operated seventy-five offices in fifty-nine countries, with twenty-five of the offices located in "key foreign cities". This amounted to a "worldwide intelligence organization" with two hundred seventy agents overseas and twelve hundred analysts in Tokyo.¹¹

The close working relationship between Japanese companies and the Japanese government is illustrated in the famous IBM-Hitachi case, sometimes referred to as JAPSCAM. In 1981 Hitachi acquired a nearly-complete set of the confidential and much coveted IBM *Adirondack Workbooks* from a former IBM employee. These were state-of-the-art design workbooks containing technical secrets clearly marked **FOR INTERNAL IBM USE ONLY**. Eventually, the combined efforts of IBM counterintelligence and FBI personnel led to the dramatic arrest of a number of IBM officials and a reported, out of court settlement of US\$300 million for IBM. What is most interesting to note, is that Hitachi spymasters in Japan, who were supervising the espionage operations, transmitted their instructions to Hitachi case officers in San Francisco through Japanese diplomatic communications. The Japanese consulate had received telex instructions on how to proceed with the acquisition program after meetings between Hitachi and the American agents had occurred in Tokyo. Once communication was received in the consulate, the message was transmitted to the Hitachi man in Silicon Valley by the commercial representative of the Japanese consulate.¹²

Similarly, in a remarkably candid interview appearing in *Bungei Shunju*, a Japanese monthly, in 1982, chairman of Fujitsu Taiyu Kobayashi described, in calculating detail, how Fujitsu also acquired information on IBM. While criticizing Hitachi for its blatant methods, he also explained how his firm avoided direct runs at IBM for information in order to avoid detection and prosecution. *The Washington Post* reportedly had planned to reprint the

Kobayashi interview in its entirety in a January 1983 issue. However, after *The Post* obtained the rights to do so from *Bungei Shunju*, Fujitsu became aware of it and pressured the monthly to cancel the agreement. The Washington public relations firm of Ruder and Finn, which represents Fujitsu explained that, "Fujitsu suggested that the complete version not be printed in the *Post*."¹³ In another case it is believed that Fujitsu placed a Japanese mole inside Fairchild Semiconductors between 1977 and 1986 which did substantial damage to the company. This may be part of the reason why Fujitsu attempted to purchase Fairchild in 1986, while they were in secret negotiations with Cray Research Corporation for closer collaborations. These few examples are merely the tip of the iceberg when it comes to Japanese industrial espionage prowess. Unfortunately, as with all intelligence operations, the best and most successful often are never publicized or become known.

4. RUSSIA'S COLD WAR CONTINUES

All governments actively seek to acquire significant military technology and equipment. The spy versus spy game has long attempted to acquire technological breakthroughs and to break adversary cryptographic codes, so as to design counter measures that could neutralize a potential adversary. However, it was not until the early 1980's that the extent of such a program by the Soviet Union was fully appreciated. In an extraordinary exchange between President Mitterand of France and US President Reagan in Ottawa, Canada, in July, 1981, the French President briefed his American counterpart on "**Farewell**." "Farewell" was the cryptonym for a high level Soviet source within the Soviet Intelligence service technological theft program.¹⁴ "Farewell" may be one of the greatest agents the West as a whole has ever run against in the Soviet Union. The man, who was prophetically code-named "Farewell," nevertheless elected to disclose to the West the entire order of battle for the massive Soviet effort to acquire Western technology. It led to a great number of Soviet KGB and GRU expulsions from France and other Western countries and produced extremely detailed intelligence into the methods of operation of the Soviet effort, their take, and targets. A more recent book, *The "Stormbirds"*, included even more detailed information.¹⁵ It is reported that over 2,000-plus secret and top secret intelligence documents and two personnel lists of secret Russian, agents which he copied directly from the card indexes of Department "T", were initially provided only to the French. This allowed the identification of many KGB Line X officers and their assets. Information was obtained which was so detailed as to include the devastating statistics that 61.5% of all technical information then collected by the Russians came from American sources. Some of that material was declassified and released to the public in 1985 and is available for review.

The trend toward the recognition of the greater role of economic intelligence in our society is displayed most evidently in a series of articles published in *Pravda* beginning in 1990. In *Pravda*, on September 16 of that year, an article appeared entitled "*In the Holy of Holies of Security - A Journalist has Crossed the Threshold of the Eighth's Main Directorate of the USSR KGB for the First Time.*" The significance of this article was embedded in the fact that it showcased one of the more interesting revelations of "glasnost". In an extremely revealing exposé, a new strategic direction was posited for the protection of commercial communications.

*"Today, with regard to the changes in the world, new tasks face the cryptographers of the KGB: The amount of work in military directions is being reduced while the work on commercial issues is growing... The need to protect commercial secrets is becoming urgent, especially more recently in connection with the movement of our departments, enterprises and cooperatives into the foreign market... Our secret service will offer all types of aid and consultations to appropriate organizations at all stages in the design, creation, introduction, and use of such cryptographic codes. This has not changed in today's Russia."*¹⁶

This extraordinary article even claimed that U.S. intelligence, in particular the National Security Agency (NSA), was actively aiding U.S. businessmen in their negotiations concerning trade in agriculture and oil. This allegation later will be referred to on in this report.

As the former East Bloc countries enter Europe and the world of economic competition, many changes will be thrust upon their societies. Unemployment and inflation are among the first indications of the transition to a free market economy. The question remains—what will they rely upon for their competitive edge? One or more elements will surely be required for success. Price, quality, technical innovation, service, etc. are examples of traditional resources which can provide a competitive advantage. However, a non-traditional edge could be provided by the national intelligence services. As noted earlier, the services set up by the Soviet KGB were

extremely competent. One of their major tasks was the clandestine acquisition of militarily-significant technology. The "Farewell" case points to these. As another example, some of the most lethal hemorrhages of technology were run by the Polish and East-German intelligence services. Certainly, privatized intelligence collection efforts on behalf of companies must also be taken into account. Former intelligence officers who were put out of work as governments changed in the aftermath of the cold war could be a source of talent for such private and corporate undertakings. But, the real question remains: Will governments task their present intelligence services, dedicated to military and national security, to gather economic intelligence in order to keep their countries competitive?

In June of this year, a new document entitled "New Approaches to National Security Problems" was obtained by news media organizations in Moscow, which might shed some light on this question.¹⁷ The report, supposedly written by the Russian Security Council before the appointment of General Aleksandr Lebed, but endorsed by him, clearly states that Russia must recreate its state's management system based on the "information highway". It also describes "economic security issues as dominant at this present stage of the country's development." To this end General Lebed is reportedly to have stated:

I believe that it is necessary to adjust the structure and goals of the intelligence services directing their efforts to back Russia's economic interests in the first place. I will demand that more effort be applied immediately in the following specific fields: to ensure an uninterrupted monitoring of the situation on the world markets of armaments, aviation and space equipment and to search for information on existing or developing technologies in the design of new armaments; to search for new designs in commercial technologies, both by state-run and private enterprises; to search for critical information on the plans and activities of the leading international financial institutions, major transnational corporations, banks and investment companies of all countries of the world; to organize information campaigns in foreign countries to attract more investment in the Russian economy. I believe it is necessary to allow the intelligence services to cooperate with major domestic production and financial enterprises of any form of property. The experience of France, Germany, Japan and China has proved the efficiency of cooperation for raising the competitiveness and technological potential of the domestic economy. These are the issues that I want to handle within the Security Council through coordinating the resources that already exist in the key financial and economic structures of the state.¹⁸

Other articles soon followed in the Russian press on the "Future of the Intelligence Community". These included more specifics about Lebed's plans to increase the number of GRU (Russian Military Intelligence) and SVR (Foreign Intelligence Service, former KGB First Chief Directorate) officers stationed around the world specializing in science and technological espionage. Countries specifically mentioned for technology collection include: United States, Great Britain, Japan, Israel, the Federal Republic of Germany, France, Italy, South Korea, and Sweden. Switzerland was also identified for intelligence personnel with higher financial education and directs the curricula of the GRU and SVR intelligence schools to be revised so as to increase the number of subjects in economics and finance. Also mentioned was the need to provide Westerners with access to classified technological information funds in the millions to stimulate growth of those willing to cooperate with the overseas intelligence posts. Lebed will supposedly take personal charge of the effort to support and form financial-industrial groups (FIGs) which might become the symbol of Russia's industrial might in the third millennium. These FIGs will be created mostly out of defense enterprises oriented toward the world market. An example of an emerging FIG is the military-industrial complex supporting the exports of MIG-29 fighter aircraft.¹⁹

5. PROTECTING THE BUSINESS OF THE UNITED STATES

The theft of U.S. proprietary information and technology by foreign companies has long been part of the competitive business environment. But, as former Director of the Central Intelligence Agency Director Gates pointed out in his testimony to the U.S. Congress in April, 1992, it is the increased activities of foreign governments in industrial espionage which is raising the level of concern: *"Some foreign intelligence services have turned from politics to economics and the United States is the prime target."* President Bush emphasized the same concern by saying that we must thwart anyone who tries to steal our technology or otherwise refuses to play by fair economic rules. Director Gates stated that *"Various governments in Asia, Europe, the Middle East, and to a lesser degree, Latin America, as well as some former Communist countries (some 20 countries or governments in all) are involved in intelligence activities that are detrimental to our economic interests at some level."*²⁰

France, Japan, and Russia are by no means the only countries which have established a formidable apparatus for gathering business intelligence. The creation of such business intelligence systems is now recognized as a good national and corporate business decision. Meyer points out that the big American grain houses and large computer companies, to mention a few, have also formed similar units. However, most U.S. companies do not collect information and process it into true intelligence. The competitive world of international trade and finances will more than likely require it as well. However, the Japanese seem to have organized themselves within this capacity and it has become an extremely effective element for their overall competitiveness and success. Much can be learned from the success of the major Japanese trading companies. Acknowledgment of the Japanese achievement in business management should now be extended to that of business intelligence. American business must respond to this challenge not with complaints about the Japanese but with constructive action. "Japan bashing" should have no place in this discussion. The Japanese, French, and the Russians are doing their best to advance their interests - and so should we.

The allegation made in *Pravda* that NSA was actively aiding U.S. businessmen in their negotiations concerning trades in agriculture and oil, and that it would be increasingly involved in economic espionage, also against the USSR, is clearly disinformation designed to justify the Russian program.²¹ If the truth would be known, there is tremendous trepidation on the part of the U.S. government to do anything this risky. The mere attempt to devise a method by which to disseminate sensitive information, to whom and for whatever reasons, is enough to stop any discussion of this topic right in its tracks. In America's open democratic system it is simply too hard to accomplish and would be contrary to the free market principles espoused by the U.S.. While these charges are clearly meant for internal consumption, it is interesting to postulate that even the KGB seems hard pressed today to justify their existence and still relatively enormous budget.

Although, industrial espionage is not new, the U.S. Federal government's response clearly appears to be changing. On August 10, 1990, FBI agents arrested Bernard Mayles after allegedly turning over micro-organisms and documents to an undercover FBI agent. This is the first major criminal espionage case involving, not classified United States defense documents, but those of a private pharmaceutical company. The undercover agent had offered the huge sum of \$10 million for the trade secrets of two of the nation's largest pharmaceutical companies. In the future, corporate spying may be getting more attention from the FBI financial crimes unit. In 1988, a National Institute of Justice survey showed that 48% of the 150 companies it polled had been victims of trade secret theft.²² A majority of those companies had been victimized more than once. Given the natural incentives of companies not to report the fact they have been victims of such crimes, common sense indicates that percentages are in fact low. However, adverse information, true or false, could send a company's stock plummeting. If the "year of the spy" taught us anything, it is that more people than we would like to admit are ready to compromise their ethics and national alliance for *money*.

American companies spend an estimated \$108 billion in R&D, or roughly one quarter of the world's investment in research. It should be no surprise, therefore, that U.S. R&D are the top target of industrial espionage. Other targets include: new technology, customer lists, program plans and financial data. The Mayles indictment indicates just how vulnerable pharmaceutical companies are. A \$10 million dollar bribe pales in comparison to the enormous amount of time and money it takes to bring a new drug to market. Trade secret experts maintain that pharmaceutical companies spend an average of 12 years and \$231 million to research and market a new drug. Herein lies the economic incentive which will drive a very active industrial espionage business. This all leads to one conclusion: American business faces many threats which in turn challenge our ability as a nation to maintain our economic competitiveness. This is a national security strategic issue since it means opportunities for economic growth. Protection of propriety information must not only be limited to foreign countries or companies, but American competitors within the United States. In my opinion, the most common threat to proprietary information is not from the electronic "bug" but from the open mouth. It is loose lips which can provide a competitor with more than is prudent. Electronic communications and eavesdropping on the national and global information highways can carry an inadvertent or willful sensitive remark near-instantaneously, and irretrievably, into unknown adversary camps.

The protection of propriety information in an increasingly competitive global market may ultimately incorporate the traditional means and procedures developed for the protection of classified government information. Corporate security may become a necessary component for survival, just as sound financial judgment and planning have traditionally been in the past.

6. INTERNATIONAL TERRORISM AS A GLOBAL THREAT

What are the new dangers to our technological society? How vulnerable is our information-dependent society as it evolves into a more efficient multi-faceted society electronically interconnected and responsive? What does the end of the cold war mean to our vulnerabilities and perceived level of threat. Do vulnerabilities and threats require an enemy to be named to survive?

International terrorism is now the emerging global concern. Specifically, in 1976 the Swedish Ministry of Defense published a report on "**The Vulnerable Computer Society.**" This report explores the threat of sabotage and the increasing vulnerability of society as a whole during wartime because of the increasing use and reliance on computers. The resulting economic vulnerability of any society is then underlined in the report with the recent problems involving telephone networks. It is noted that repeated interruption of communications can produce powerful psychological warfare effects on the more technologically advanced societies. Especially affected are societies such as ours which have little, if any preparation to this type of disaster and psychological warfare.

In February of 1989, a series of hearings were held by the United States Committee on Government affairs. Experts from the Office of Technological Assessment who testified indicated that most electrical systems in the United States were not prepared for the advent of multiple serious failures to the system, whether they be inflicted by natural disaster or by from a terrorist attack. The results of Hurricane Hugo confirm this. A systematic attack by terrorists upon multiple key electric power plants and their distribution centers, and destruction of the telephone switching stations, could have effects which would last for extended periods of time. Public utility companies just don't seem to conduct serious disaster planning that take into account the simultaneous knockout of multiple facilities. Equipment is not stored or propositioned to be rapidly set up in anticipation of an emergency of this magnitude. And even more surprising is the fact that many key installations do not even offer a moderate level of defensive precautions. To wit, when the high-capacity transformer which provides electric power for the Lawrence Livermore National Laboratory (LLNL) was destroyed by a series of escalating accidental events, a replacement transformer was not readily available. It took several months before the replacement could be found and transported to from the East coast to California to restore electric power at full load.

Another interesting report emerged from those 1989 hearing for the Senate Governmental Affairs Committee. The report was provided by one of our intelligence agencies. I quote from their public statement:

"Virtually all of today's computer systems and networks are susceptible to virus attacks. Protection techniques vary depending on the type of computers and networks in use. Currently, defenses against computer viruses fall into two categories: procedural and technical. Procedural protection consists of those actions taken to restrict access to the system or network. Personnel security, physical security, and administrative security play a major role. Technical defenses are software and/or hardware solutions that prevent, detect, or confine viruses. Because we lack effective technical defenses, we rely heavily on procedural protection. Strict adherence to good computer security procedures and policies is one way to defend against introduction of a virus. However, to provide increased protection, additional research is required to further develop effective technical defenses."

7. THE INVISIBLE WEAPONS of ELECTRONIC WARFARE

Now that we have left behind the cold war as a period of high tensions and *high stability*, how vulnerable are we? Have we adjusted psychologically and militarily to a world of low tensions but *high economic instability*? Is the world now a more dangerous place due to unprovoked terrorism and crime that can strike any place? Information Warfare is another area which has captured much of the discussion concerning the changing face of war. In Russia studies of the American effort during the Gulf war have lead to studies and formulations of informatic warfare as the wave of the future. The Rand corporation recent publication of Strategic Information Warfare summarizes its research conducted for the Office of the Assistant Secretary of Defense (Command, Control, Communications and Intelligence) on this extraordinary subject. These areas continue to outstrip the imagination as one ponders the ongoing rapid evolution of cyberspace and the information culture of our advanced post industrial society.

Until now, most attacks on computers by terrorists have not been by a hostile virus but with explosive bombs. Let me explain: The EDP Auditors Association of Israel has monitored terrorist attacks on data processing (DP) personnel and facilities since 1969. Between 1969 and 1981 they reported fifty-four documented cases of direct DP terrorism. Twenty-two attacks were directed at United States companies abroad, half of the fifty-four cases took place in Italy and were the work of the **'Red Brigades'** and other terrorist groups within their network. The Red Brigades clearly state in their own manifesto that computer targets are a key element of the capitalist technological society and that their vulnerabilities permit devastating destruction. Euro-terrorism has been in the forefront of attacks against computer facilities and companies. In the United States, our technological infrastructure is more vulnerable to a variety of such attacks—both those which are obviously malicious, and those which, while still benign, can be devastatingly destructive. For example: Robert Tappan Morris who unleashed the **WORM** on the Arpanet system caused a tremendous problem when the speed of the network rapidly replicated the **WORM** and it shut down much of the nation's digital communications. The government estimated the cost of the Morris experiment gone wrong at about \$160,000. Kevin Metnick, the phone freak depicted in **CYBERPUNK**, is the epitome of the public's dark-sided computer hacker. This case is only an indication of what is possible by those harm bent on harm could accomplish.

The Middle East certainly has developed an image for originating some nasty computer viruses. The **IDF** virus, named after the Israeli Defense Force which identified it, is just one example. Another is the **PLO** virus, and then there is the **May 29th** virus. These examples point to the vulnerability of computer systems and the prevailing methods of disabling attacks with ingenious viruses as agents of warfare. The result is a relatively low-intensity economic and psychological aggression. In one of the more turbulent areas of the world it could be played out as follows: Israel is a society which is known both for its technology as well as existing in a hostile ethnic environment. The clandestine placement or activation of a computer virus within Israel by adversaries could become a critical impediment to wartime response and even bring into question survival. A **"logic bomb"** implanted within a target computer could become activated upon program change. A **"time bomb"** could launch a virus on a specific date. Strategic placement of viruses, can have a devastating or fatal effect. Such adversary actions must be of prime concern to the security professional because of their devious potential nature. Unlike these hostile viruses, the computer virus called **"Brain"** had the benign intent to catch computer software pirates, but because it proliferated without bounds many innocent computers were infected and their files destroyed. This virus was developed by two Pakistani brothers and could be viewed as another example of a computer prowler limited to the near East. But it is an omen of impending threat to our society that is much more vulnerable because of its growing dependence upon computers.

In France, new department within the DST has been created. This is the **'DST Protection Department.'** It will attempt to survey, monitor, and provide security for uses of the **INTERNET** by French business. The original idea was for it to somehow control the use of the **INTERNET** in France, but it soon became clear that this was an unrealistic goal. The concern over the **INTERNET** as an uncontrollable, international network was generated after the 1988 crash of the **Airbus**. A French company prepared a confidential internal report on the crash, but soon after it was completed, the most damaging portions of the report, citing design flaws, was distributed freely on the **INTERNET** through a Finnish institute. The French company claims that this leak was fostered by the U.S. competitor to **Airbus**—Boeing.

However, the conscious placement of a virus may also have positive applications. In the August 1995 issue of **Armed Forces Journal**, an article appeared with the following title: **"Virus" implants "Go-Signals" in Future US Weapons?** The article describes how a virus or software disabling mechanism could be placed in every major new US weapons system—one which could be remotely triggered if the system, and the know-how for its activation, fell into an enemy's hands, or come under adversary control—could determined the outcome of the war without a shot being fired. While the article can be described as speculative at best, the idea of placing a radio receiver, embedded among thousands of other circuits on a key microchip to disable the system is both novel and intriguing, provided that the knowledge about its existence could be protected. The other side of the equation is, what could occur if an enemy was to obtain such a code or ability to activate the disabling program or redirect the compromised weapons upon their original owners. The answer is obvious and troubling; the nation's newest and most innovative weapons systems could be neutralized.

Information Warfare (IW) represents a growing concern among military war planners and Washington policymakers. Cyberspace, with all its promise also represents a growing set of uncharted vulnerabilities and in turn opportunities for both friends and foes. As we move forward as a society and a government which more and more intertwining telephony and computing, the need to better understand and protect the software links between these technologies becomes more and more challenging. As the world become better connected via the Internet and other wireless ways, the ability to reach out and touch a potential adversary increases exponentially. For these reason the United States government, as well as many others around the globe are exploring this new element of warfare. A recent RAND report conducted for the Department of Defense immediately comes to mind, for it moves our understanding from the tactical applications of IW techniques to the strategic level.

The United States has substantial information-based resources, including complex management systems and infrastructures, involving the control of electric power, money flow, air traffic, oil and gas, and other information dependent items. U.S. allies and potential coalition partners are similarly increasingly dependent on various information infrastructures. Consequently, if and when potential adversaries attempt to damage these systems using IW techniques, information warfare inevitably takes on a strategic aspect.²³

Some of the features of Information Warfare (IW) make the threat difficult to truly evaluate. Seven features were identified in the RAND exercise. They include: low entry cost; blurred traditional boundaries, public vs. private interests, criminal or warlike behavior, geographic, etc.; expanded role of perception management; new strategic intelligence challenge; formidable tactical warning and attack assessment problems; difficulty of building and sustaining coalitions; vulnerability of the U.S. homeland. These challenges will require sustained and creative government responses for many years, with the ever changing technological telecommunications and information environments.

8. CONCLUSION

In conclusion, the real issue for the business community is not to construct operatives and organs for clandestine operations and collection, but systems that could help us organize and visualize information in new ways. Business should not expect the United States government to do their job for them, or act as an agent on their behalf for competitive advantages. However, if foreign governments are in fact aiding their own domestic companies, an American policy could address this issue diplomatically and politically. For many businesses much of the information is already being collected by trade organizations or is otherwise within easy grasp. However, the systematic approach and coordination of such effort can transform the bulk of information gathering into succinct information Intelligence. A corporate Intelligence office, operating as an independent unit of the corporate strategic planning group, could transform an organization's traditional approach to Information Into a new and powerful business tool. Its formulation will no doubt take many reiterations, but the emergence of Intelligence components will surely continue to grow in the business community.

For some businesses, It may suffice to bring to the same table those individuals who had collected Information for the company in the past, but had perhaps not worked together In an organized and coherent manner. For others, It will be the establishment of a small group of employees who will collect, synthesize and feed to the top a new and powerful decision making tool. Whether it is called intelligence, or strategic planning, the new business management tool will emerge and increase In importance. Competition will force its existence to become commonplace.

A cogent example of such corporate activity has been practiced and demonstrated by the Phillips Company in Aachen, Germany. There, several hundred lawyers and technical professionals jointly review the global intelligence collected from their field offices, appraise the technical literature, and analyze global patents for their significance to Phillips' strategic business plan. This organization of legal and technical experts, working in part on a rotating basis by assignment from other corporate departments, has been successful not only by advising corporate decision makers, but also by filing a growing number of preemptive patents for Phillips to assure it a leading position for years to come.²⁴

It should be obvious by now that we must plan to protect our domestic infrastructure and national resources against an information warfare attack and industrial espionage conducted against our

economic and technological base. In some cases we must consider this in alliance with other nations, in others not. The time to establish a viable policy to operate our information highways as safe conduits for electronic business is overdue. The effects such hostile actions could have on companies and our economy would have devastating implications to our inter-dependent global economy. Once certain forces are unleashed, even if in a limited capacity, the consequences are far from predictable and the potential damage is strategic.

REFERENCES

- ¹ *U.S. News & World Report*, June 3, 1991, "The New Spy Wars", p. 23.
- ² *The Washington Post*, November 6, 1991, Business section.
- ³ Herbert E. Meyer, *Real World Intelligence*, New York, 1987, p.8.
- ⁴ *Ibid*, pp. 8-9.
- ⁵ Meyer, Op. Cit.
- ⁶ Strategic Information Warfare, The Rand Corporation, 1996
- ⁷ *The New York Times*, July 1, 1996, Page one, "Economic Summit Subplot: Do French Walls Have Ears?", David E. Sanger
- ⁸ Bob Woodward, Veil, Simon and Schuster, New York, 1987, pp.368-371.
- ⁹ Pat Choate, Agents of Influence: How Japan's Lobbyists in the United States Manipulate America's Political and Economic System, Alfred A. Knopf, New York, 1990, p.37, from Meyer, Op Cit. p. 58.
- ¹⁰ Peter Schweizer, Friendly Spies : How America's Allies Are Using Economic Espionage To Steal Our Secrets, New York, The Atlantic Monthly Press, 1993, pp.71-72. Admiral Pierre Lacoste of the French Intelligence Service further states that MITI and JETRO intelligence operations are massive, sending five hundred thousand messages back to Tokyo each day.
- ¹¹ IBID. p. 80.
- ¹² Op. Cit. p.56.
- ¹³ Op. Cit. p. 68.
- ¹⁴ Thierry Walton, Le KGB en France, Grasset, Paris, 1986, Part 5, The "Farewell" Dossier Scientific and Technological Espionage, p. 241.
- ¹⁵ Gordon Brook-Shepherd, The Storm Birds, Widenfield & Nicolson, New York, 1989, Chapter 17: "Farewell" A French Connection.
- ¹⁶ Foreign Broadcast Information Service, 24 September 1990 (FBIS-SOV-90-185), pp. 64-68, Endfield.
- ¹⁷ INTERFAX, Moscow, 26 June 1996; KOMMERSANT-DAILY, (in Russian) 26 June 1996, page 3; ITAR-TASS, Moscow 27 June 1996.
- ¹⁸ "Lebed's National Security Document", INTERFAX, Moscow, June 26, 1996, 1852 GMT
- ¹⁹ Nezavisimaya Gazeta (Independent Newspaper), 1 July 1996, page 2. (in Russian)
- ²⁰ Hearings before the Subcommittee on Economic and Commercial Law, Committee on the Judiciary, House of Representatives, "The Threat of Foreign Economic Espionage to Corporations," April 29 & May 7, 1992, Senate # 65.
- ²¹ Reference 10, p. 67.
- ²² This paragraph was excerpted from a reproduction of an article dated September 16, 1990, The Business section of "The Record". This article also provided the information from the National Institute for Justice survey and the data on the U.S. spending in R&D.
- ²³ RAND corporation, Strategic Information Warfare, Washington, 1996.
- ²⁴ Personal communications with Dr. Herbert Newkirk, a chemist at the Lawrence Livermore National Laboratory who spent two years as a guest of Phillips on a sabbatical assignment.

**"B IS FOR BUSINESS : MANDATORY SECURITY CRITERIA AND THE
OECD GUIDELINES FOR INFORMATION SYSTEMS SECURITY"**

Professor William J Caelli
Head
School of Data Communications
and
Member
Information Security Research Centre (ISRC)
School of Data Communications
Faculty of Information Technology
Queensland University of Technology
GPO Box 2434
BRISBANE, QUEENSLAND, AUSTRALIA
Tel: +61-7-3864 2752
Fax: +61-7-3221 2384
Email: caelli@fit.qut.edu.au

Paper submitted to the
19th National Information Systems Security Conference,
Baltimore, MD, USA, October 22-25, 1996.

Abstract:

This paper sets out the proposition that *mandatory* security functionality, with its associated enforcement and evaluation criteria, are required in computer and data network systems to meet emerging national and international laws and guidelines for information systems security. The OECD 1992 Guidelines for Information Systems Security are used as a baseline for the consideration of such levels of trusted functionality. Concepts for trusted computer and data network systems, as set out in the original Trusted Computer System Evaluation Criteria (TCSEC) of the United States, the Information Technology Security Evaluation Criteria (ITSEC) of the group of four European nations, the Canadian (CTCPEC) evaluation criteria and the more recent international Common Criteria (CC) are seen as relevant to the distributed and client/server computing environments of information systems in the 1990s and beyond. Overall, it is suggested that security functionality and evaluation/enforcement, at the level of the earlier TCSEC "B1" as a minimum, are required in networked computer systems to meet emerging national and international legal requirements and I.T. security guidelines.

Keywords:

mandatory computer security, OECD security guidelines, TCSEC, ITSEC, Orange Book.

"B IS FOR BUSINESS : MANDATORY SECURITY CRITERIA AND THE OECD GUIDELINES FOR INFORMATION SYSTEMS SECURITY"

1. Introduction - TCSEC/ITSEC AND THE OECD Security Principles, 1992

1.1 Introduction

In 1992 the Organisation for Economic Co-operation and Development, or OECD, [OECD-92] created a set of "Guidelines for the Security of Information Systems". These guidelines follow an earlier set in 1980 covering the provision of privacy in relation to stored and transmitted data [OECD-80]. Together these guideline documents have had a marked affect on the development of associated legislation and standards worldwide at the national, regional and international levels. This paper argues that in order to meet the managerial level requirements set out in these guideline documents, information technology (IT) professionals, owners and users of information systems and appointed managers of information systems must move to a "mandatory" concept of system security facilities in order to develop and operate information systems that comply with the guideline requirements.

The key theme of this paper is that **mandatory access control**, as set out in the original "Orange Book", the Trusted Computer System Evaluation Criteria or TCSEC [TCSE-83] and later computer security evaluation criteria such as the Information Technology Security Evaluation Criteria or ITSEC [ITSE-90] and the Common Criteria or CC [CC-96] represent a minimal requirement for protection in distributed computer systems linked via data networks and operating under a "client-server" paradigm for applications. This argument is made on the basis of emerging national and international legislation and guidelines covering both **privacy** and **information systems security** which place mandatory obligations on users, owners and developers of information systems to safeguard the information entrusted to them. It is argued that lower level **discretionary** security is insufficient to provide reasonable security assurances in interconnected systems, particularly where such activities as acceptance of "scripted" programs by a host computer from a remote system may be allowed. The provision of mandatory security services will also form a basis for what may be called "self-defending objects", a scheme whereby distributed object oriented systems may make intelligent security related decisions about requests made to them from remote and often unknown sites.

1.2 The OECD Guidelines

On the 26th of November 1992 the Council of the OECD adopted a document known as the:

"Recommendation of the Council Concerning Guidelines for the security of Information Systems".

These were then adopted by the 24 member nations of the OECD. The document of interest here is composed of three parts, consisting of the "Recommendation" above plus:

"Guidelines for the Security of Information Systems"; and

"Explanatory Memorandum to Accompany the Guidelines for the Security of Information Systems".

The "Guidelines" themselves set out **nine principles** for security which are set out in Section 2. The "Recommendation" sets out the responses that member nations should have to the guidelines document. In particular it states that member countries should:

1. *establish measures, practices and procedures to reflect the principles.....*
2. *consult, co-ordinate and co-operate in the implementation of the Guidelines, including international collaboration to develop compatible standards, measures, practices and procedures*
3. *agree as expeditiously as possible on specific initiatives for the application of the Guidelines,*

4. *disseminate extensively the principles...*
5. *review the Guidelines every five years with a view to improving international co-operation on issues relating to the security of information systems."*

An important consideration has to be one of the response of IT professionals worldwide, through their professional organisations and through their international body, the International Federation for Information Processing (IFIP), to these Guidelines and the associated Recommendations. It is likely that legally binding parameters may emerge in the 1990s that govern the responsibilities of IT professionals, information systems owners/users and system managers alike. In this sense the underlying technology that enables security to be incorporated into information systems becomes critically important. Thus, consideration of any technical guidelines and parameters for judging such security becomes important since these, along with the Guidelines, could become "base-lines" or "codes of minimal acceptable practice" by which IT professionals, in particular, may be judged. Such judgements may even take the form of legally binding decisions through the judicial system as well as being a more general form of assessment adopted by society at large.

The actual Guidelines themselves are "*addressed to the public and private sectors*" and apply "*to all information systems*". Moreover, the scope of the guidelines is one based around the generally accepted definition of security, i.e. the confidentiality, integrity and availability of information systems. The Guidelines also have a set of six stated "Intentions" of which one, "... to foster confidence in information systems and the manner in which they are provided and used..." is pertinent to this paper. In summary, it is submitted that such "confidence" can only really be generated in information systems, that now consist of connected host systems on national and international data networks, by incorporation of so-called "mandatory" computer security technology in the base computer systems and data networks themselves.

1.3 National and International Legislation and Guidelines

Following on from the OECD Guidelines there has been a number of initiatives at the national level, e.g. in the United Kingdom, Australia, and elsewhere, to give greater force to these guidelines in line with the recommendations of the OECD. The "Code of Practice for Information Security Management" [BSI-93] of the United Kingdom, now a British Standards Institute standard, BS-7799, sets out specific requirements that may be:

".... used as a common reference standard for inter-company trading and for sub-contracting or procurement of information technology (IT) services or products."

It goes on further to state that:

"... Information security threats are expected to become more widespread, more ambitious and increasingly more sophisticated."

This code clearly sets out parameters that could be reasonably determined to be minimal "statements of due care" in relation to the responsibilities of IT professionals and system managers. In this sense, these codes and guidelines may start to take on some form of legal force when offered as guides in any legal proceedings.

These documents are starting to set a scene under which computer and data network security will increasingly become the responsibility of IT professionals and managers in a legally binding sense. At the same time, then, these IT professionals and managers will need to be sure that the products and systems used to create an overall enterprise wide and cross-enterprise information system are "safe to use" for such needs.

1.4 Evaluation Criteria

The USA's National Research Council [NRC-91] described security evaluation criteria in the following terms:

"At a minimum, security evaluation criteria provide a standard language for expressing security characteristics and establish an objective basis for evaluating a product relative to these characteristics..."

This clearly identifies security evaluation criteria as relevant when considering the OECD guidelines.

The earliest attempts at creating such documents belongs essentially to the United States Department of Defense [TCSE-83] and the resulting "Trusted Computer System Evaluation Criteria" or TCSEC. Work had started in the late 1970s under both the United States Department of Defense (DoD) and the National Bureau of Standards (NBS), with the assistance of the Mitre Corporation, to create base documents that addressed computer security issues, e.g. Ware, 1979 [WARE-79].

**Department of Defense
Trusted Computer System Evaluation Criteria
15 August 1983.**

Division C: Discretionary Protection.

Classes in this division provide for discretionary (need-to-know) protection and, through the inclusion of audit capabilities, for accountability of subjects and the actions they initiate.

Division B: Mandatory Protection

The notion of a TCB that preserves the integrity of sensitivity labels and uses them to enforce a set of mandatory access control rules is a major requirement in this division. Systems in this division must carry the sensitivity labels with major data structures in the system. The system developer also provides the security policy model on which the TCB is based and furnishes a specification of the TCB. Evidence must be provided to demonstrate that the reference monitor concept has been implemented.

Division A: Verified Protection

This division is characterised by the use of formal security verification methods to assure that the mandatory and discretionary security controls employed in the system can effectively protect classified or other sensitive information stored or processed by the system. Extensive documentation is required to demonstrate that the TCB meets the security requirements in all aspects of design, development and implementation.

USA Department of Defense TCSEC - 1983.

The resultant document was the 1983 TCSEC document, updated in 1985. The TCSEC sets out three fundamental security requirements summarised as *policy*, *accountability* and *assurance*. These requirements were combined into a set of security "divisions" and then further into "classes" within these divisions which would characterise computer system security in a succinct manner, as shown in the previous table.

Towards the late 1980s, however, these criteria were being augmented by other criteria developed by other nations, e.g. Canada, United Kingdom, Germany, The Netherlands, etc. for a number of technical and political reasons. In particular, 1990 saw the convergence of the work in France, Germany, the Netherlands and the United Kingdom into a set of "harmonised" criteria; the so-called Information Technology Security Evaluation Criteria or ITSEC [ITSE-90]. These were seen as a superset of the earlier TCSEC and expanded the concepts to emerging new IT products and systems. The main conceptual advances in the ITSEC were the:

- separation of the concepts of security "functionality" and security "evaluation" into distinct categories; and
- coverage of both "products" and "systems" in a single document set.

The evaluation or assurance criteria are set out as six separate criteria labelled E1 to E6. It should be noted that these evaluation level criteria are built into the divisions and classes of the earlier TCSEC. The ITSEC functionality criteria are likewise mapped into separate groups that firstly map the functionality of the TCSEC, called F-C1 to F-A1, while allowing for special functionality classes of high integrity (F-IN), data confidentiality (F-DC), etc. In ITSEC senses this paper examines the proposal that information products and systems need to meet a level of F-B1, E3 as a minimum to abide by the OECD and like guidelines or an evaluation level of EAL-4 in Common Criteria terminology.

1.5 Access Control/Mandatory versus Discretionary versus None

Gasser [GASS-88] set out, in his 1988 book "Building a Secure Computer System", some background for the underlying reasoning in this paper. He states:

"Until the early 1970s, it was not generally realised that two fundamentally different types of access control exist. Discretionary access control is the most common: users, at their discretion, can specify to the system who can access their files. ... Under nondiscretionary or mandatory access control, users and files have fixed security attributes that are used by the system to determine whether a user can access a file. The mandatory security attributes are assigned administratively (such as by a person called the security administrator) or automatically by the operating system, according to strict rules. The attributes cannot be modified by users or their programs."

These concepts may be extended to object-oriented concepts whereby information technology professionals may associate "methods" with *objects* or *classes* whereby those methods must be compulsorily invoked whenever an object is referenced. However, as in the case above, the question as to who has responsibility for the determination of the rules for such methods again fits into the discretionary versus mandatory debate.

There has been suggestion that the early TCSEC discretionary "C2" class of assurance is good enough for information systems in the 1990s, particularly in the banking and finance, health care and commercial government systems area. Indeed Gasser [GASS-88] points out that:

"In practice, mandatory controls do provide a benefit over discretionary controls, even if Trojan horses are not a threat, in cases of accident or irresponsibility." [GASS-88. Pg. 62].

With the introduction of so-called "scripting languages", such as "JAVA", whereby programs may be transmitted over national and international networks for execution on willing host systems, the Trojan Horse threat has taken on a new significance. This again adds force to the argument that discretionary levels of security are now obsolete and moves to mandatory levels, essential since now a "user" by definition, may be any "applet" provider. In the JAVA case, at least, the JAVA language interpreter, usually integrated into a World-Wide-Web or Internet access program or "browser", must be guaranteed to enforce the strong "typing" requirements of the JAVA language, including restrictions to local data/input-output operations.

2. The OECD Principles

The OECD has detailed a set of nine "principles" which underpin the overall guidelines set. These are labelled as:

1. Accountability;
2. Awareness;
3. Ethics;
4. Multidisciplinary;
5. Proportionality;
6. Integration;
7. Timeliness;
8. Reassessment; and
9. Democracy

"Principles". Each principle is considered below in relation to overall computer security requirements and the "mandatory" security theme.

2.1 Accountability

The essential part of this OECD requirement, and its highlight, is the requirement for the overall responsibilities to be explicit. This covers the "owners,

The responsibilities and accountability of owners, providers and users of information systems and other parties concerned with the security of information systems should be explicit.

OECD Accountability Principle

providers and users" of the system. Such a set of requirements covering the people involved with an information system can only be met by the clear definition of central responsibility for the information resources under consideration. The "responsibilities" that are to be explicit need to be clearly defined, explained and "*apportioned*" as required. In turn these need to be enforced and accountability maintained. This requires the provision of a system wide audit facility to enable any form of accountability to be effective.

Now, the TCSEC Division C classes allow for "*discretionary (need-to-know) protection*" as well as some audit facilities for accountability of subjects in the system. However, for the system security to be explicit there needs to be a recognised system security manager capable of defining and enforcing such responsibilities and accountability. This is not possible at the discretionary level where individual users are responsible for the implementation of any security paradigms on an individual program and/or data files basis. The ability to process a JAVA applet, the "product" of an unknown user is relevant here. At the TCSEC "C" level no system manager can make reasonable pre-evaluations of code. In particular, the ITSEC F-B1 functionality class enhances this requirement in relation to "channels" of communication between subjects, based on the earlier TCSEC "B1" class. If a channel allows for information of varying security requirements to be transmitted and received, in the same computer or across a network, then:

... it shall be ensured by the communications protocol that the recipient can completely and unambiguously reconstruct and pair the received data and attributes." [ITSE-91]

This condition is simply not referenced at the TCSEC "C2" or lower class levels and appears as a necessary requirement in the decade of data networks to meet OECD Guidelines. Even within a single host.

2.2 Awareness

Essentially this OECD principle requires that the overall security policy and its enforcement procedures and techniques be made known to "owners and providers". At the "Discretionary" division of TCSEC

In order to foster confidence in information systems, owners, providers and users of information systems and other parties should readily be able, consistent with maintaining security, to gain appropriate knowledge of and be informed about the existence and general extent of measures, practices and procedures for the security of information systems.

OECD Awareness Principle

such awareness may not be possible since, essentially, each individual program and data base sub-system may be separately controlled by different groups and be subject to differing security principles.

Data and programs, at the "C" level, do not need to be "labelled" with system wide parameters. Moreover, in a distributed computing environment, with client/server programming systems involved and distributed object models of information system management invoked, it is impossible for individual sub-system "owners", usually the IT professional who developed the application, to understand and disseminate all security parameters to users of their sub-system. With the development of so-called "object request brokers (ORB)", whereby distributed information "objects" may communicate in an organised manner, it would appear that such security information needs to be incorporated into the ORB in such a manner that the ORB is itself protected from misuse and "tampering". In other words, this principle extends beyond the simple documentation of security practices in such documents as an "enterprise security manual" or the like, to the actual incorporation of security parameters into the information system itself.

2.3 Ethics

The rights and legitimate interests of others simply means that system users need to be identified to the information system and global security parameters set. These parameters should be regarded as being "system" or "enterprise" wide and not just associated with individual applications or sub-systems that operate on an information system. In particular, the overall information system may need to be consistent with any national or regional laws affecting the privacy of individuals and the responsibilities for "data protection" that may exist. Such laws and guidelines are, in a computing sense, a set of "global rules" that must exist in the overall system itself. It is infeasible to incorporate them individually into every separate application that operates on the information system, particularly in a distributed system. This has significance when "scripting languages", e.g. "JAVA", Telescript, etc. are used to dynamically create applications that may be transmitted over a data network for execution on a remote host system with resultant data/information re-transmitted to the originator of the "script". The nature of such "scripts" ("applets") cannot be predicted by IT professionals at the time that an information system is created and thus appropriate security parameters must be set that are global in nature and which meet the ethics principle. Mandatory security, with appropriate enforced labelling of data and programs, is the only technology capable of meeting this need.

Information systems and the security of information systems should be provided and used in such a manner that the rights and legitimate interests of others are respected.

OECD Ethics Principle

2.4 Multidisciplinary

Once it is agreed that diverse viewpoints in relationship to an enterprise's information system must be taken into account, the need for a global information security policy that is enforced becomes essential. Any resultant "measures, practices and procedures" need to be reliably enforced over the whole system requiring that mandatory security features be provided in the system to allow for such parameters to be centrally defined and monitored. This essentially rules out the use of "discretionary" systems as each application group operates "on its own" without consideration, in principal, for other groups. It is the responsibility of the C class system to "isolate" such groups and application sets. The "mandatory" scheme enforces a multidisciplinary approach on system security management.

Measures, practices and procedures for the security of information systems should take account of and address all relevant considerations and viewpoints, including technical, administrative, organisational, operational, commercial, educational and legal.

OECD Multidisciplinary Principle

2.5 Proportionality

This principle implies that IT professionals, in creating an information system, have performed appropriate levels of risk analysis and assessment prior to placing the system in operation. While this may be uncommon such analysis is the only way to determine the level of security technology needed to provide adequate safeguards for the system. There has been a general opinion that computer systems that implement "mandatory" access control and like services are too expensive in terms of cost and resource requirements with associated degradation of overall system performance. This can now be disputed with a larger number of systems being evaluated at the so-called "B1" level of trust according to the "Orange Book" while demonstrating minimal performance degradation, e.g. Secure UNIX SVR4.1, etc. Arguments on the basis of cost against use of mandatory, trusted technology are becoming less tenable as systems generally move to this level. There is, however, a problem at the personal computer/workstation level with mass/commodity system software. Incorporation of add-in security technology to raise the level of these systems to "B" (mandatory) could be a problem, particularly where such systems may be incorporated into an enterprise or cross enterprise distributed information system.

Security levels, costs, measures, practices and procedures should be appropriate and proportionate to the value of and degree of reliance on the information systems and to the severity, probability and extent of potential harm, as the requirements for security vary depending upon the particular information systems.

OECD Proportionality Principle

2.6 Integration

This principle gives a clear direction towards overall mandatory security of information systems. In a discretionary system it is not possible to coordinate overall security parameters under the control of a security manager who can be responsible for such coordination and integration. If IT professionals develop and implement their own security schemes on an individual sub-system basis, it would appear to be impossible to create a coherent system of security even if security

Measures, practices and procedures for the security of information systems should be co-ordinated and integrated with each other and with other measures, practices and procedures of the organisation so as to create a coherent system of security.

OECD Integration Principal

parameters for an enterprise are clearly set out in appropriate system development documentation. In particular, levels of enforcement may vary across individual application sub-systems.

2.7 Timeliness

In many cases, computer and data network security systems aim at the prevention of security related events that may compromise the overall system. There has been growing interest in the problems of recovery after a security event has occurred and the incorporation of such recovery technologies and procedures into information system security schemes. With a mandatory

philosophy such recovery facilities can be centralised and controlled whereas with any security level below this individual security recovery processes may need to be taken for each and every application sub-system that operates within an overall information system. This could be a major problem in a distributed system where individual host computers may be physically separated and be under the control of different management group in an enterprise or across co-operating enterprises, such as in the case of electronic data interchange (EDI) schemes.

Public and private parties, at both national and international levels, should act in a timely co-ordinated manner to prevent and to respond to breaches of security on information systems.

OECD Timeliness Principle

2.8 Reassessment

Periodic assessment of overall information system security becomes only feasible with centralised systems of security management and enforcement. Moreover, if changes are needed as a result of such reassessment then it is totally impractical to mandate changes to all application level programs and data structures at the discretionary level of system architecture. Centralised security features and

their enforcement dictate the use of mandatory security services at the operating system level for all hosts in a distributed computing network, regarded as the norm for information systems into the 21st century. This does, however, mean that research is needed into the dissemination of such changes in security parameters between trusted hosts in a computer network to mirror security changes in individual mandatory access control and system security schemes in connected computer hosts.

The security of information systems should be reassessed periodically, as information systems and the requirements for their security vary over time.

OECD Reassessment Principle

2.9 Democracy

This requirements could be best considered in relation to the reassessment principle. Overall system security requirements must be measured against legitimate legal rights in a democratic society. For this to be possible, overall responsibility for system security management must be identifiable and obvious. This means that if security parameters are left to individual developers of sub-systems and applications in the information system it may become impossible for this principle to be implemented and such implementation to be checked in real system cases. Mandatory information system security schemes could assist in implementation and management of this principle.

3. Conclusions

There is, however, a problem. This paper has argued that the mandatory (B level functionality) specifications of the TCSEC and ITSEC provide a base for definition of "commercial-level" functionality classes that meet the needs of emerging security guidelines and legislation internationally at the levels of F-B1, E3 for ITSEC and B1 for TCSEC and equivalent Common Criteria levels. By contrast the probability that manufacturers of computer

The security of information systems should be compatible with the legitimate use and flow of data and information in a democratic society.

OECD Democracy Principle

hardware, system software and "generic" application systems, as well as necessary intermediate software systems such as network protocol sets, graphical user interfaces, etc., will embrace such security and quality features soon, is very low. This was alluded to in the 1991 report of the United States National Research Council (NRC) entitled "Computers at Risk" [NRC-91], Page 145, as follows:

"The slow growth of the market for secure software and systems feeds vendors perceptions that its profitability is limited. Both high development costs and a perceived small market have made secure software and systems development appear as a significant risk to vendors. Moreover, a vendor that introduces a secure product before its competitors has only a year or two to charge a premium. After that, consumers come to expect that the new attributed will be part of the standard product offering. Thus the pace of change and competition in the overall market for computer technology may be inimical to security, subordinating security-relevant quality to creativity, functionality, and timely releases or upgrades. These other attributes are rewarded in the marketplace and more easily understood by consumers and even software developers."

However, the problems of incorporation of safety features and the reliability of those features into products and systems has long been recognised in other industries such as the car industry, fire prevention sector, etc. Car manufacturers did not incorporate seat belts in cars as a standard offering until it became mandatory under law. Office building proprietors did not include fire extinguishers and sprinkler systems until, again, it became compulsory by law. There is no reason to believe that the computer and telecommunications industries are any different, as has been indicated by the NRC report above. Even consumers themselves normally do not consider safety and security unless compelled to do so, at least beyond fundamental and basic levels, e.g. door locks, car locks, etc. as evidenced by the lack of penetration of smoke detectors in buildings, homes, etc.

The OECD Guidelines and any associated national responses to them, in the form of computer and data network security legislation, could assist in changing the scene, as it did for the car industry. The mandatory security features of the TCSEC and ITSEC and, in particular, a rework of these for commercial level requirements under the Common Criteria, could be incorporated into "mainstream" computer systems, particularly distributed systems. These could then be sent to meet growing world requirements on management to comply with the OECD Guidelines and associated national developments of these.

4. References

- BSI-93 British Standards Institute, 1993.
"A Code of Practice for Information Security Management"
ISBN 0 580 22536 4.
- CC-96 Common Criteria, Vers. 1.0.
Common Criteria Implementation Board (CCIB),
<http://csrc.nist.gov/nistpubs/cc>
- CLAR-87 Clark, D.D. and Wilson, D.R.
"A Comparison of Commercial and Military Computer Security Policies"
in Proceedings of the 1987 IEEE Symposium on Security and Privacy,
Pg. 184-195
IEEE Computer Society, USA, 1987.
- GASS-88 Gasser, M.
"Building a Secure Computer System"
Van Nostrand Reinhold Company, New York, USA, 1988.
ISBN 0-442-23022-2.
- ITSE-91 Information Technology Security Evaluation Criteria (ITSEC),
Version 1.2, 1991
United Kingdom, Germany, France, The Netherlands.
- LIPN-82 Lipner, S.
"Non-discretionary Controls for Commercial Applications"
in Proceedings of the IEEE 1982 Symposium on Security and Privacy"
IEEE Computer Society, 1982.
- NRC-91 National Research Council, U.S.A.
"Computers at Risk : Safe Computing in the Information Age"
National Academy Press, U.S.A., 1991
ISBN 0-309-04388-3.
- OECD-80 Organisation for Economic Co-operation and Development (OECD)
"Guidelines on the Protection of Privacy and Transborder Flows of Personal Data",
23 September 1980.
OECD 1981.
- OECD-92 Organisation for Economic Co-operation and Development (OECD)
"Guidelines for the Security of Information Systems", 26 November 1992
Document : OECD/GD(92)190.
- WARE-79 Ware, W. (Editor)
"Security Controls for Computer Systems : Report of Defense Science Board Task
Force on Computer Security"
R-609-1, Reissued October 1979
Rand Corp., Santa Monica, Ca., USA, 1979.

MARKETING & IMPLEMENTING COMPUTER SECURITY

Mark Wilson
National Institute of Standards & Technology
Building 820, Room 426
Gaithersburg, MD 20899

I had spent several years in a previous job trying to convince computer system users and managers that they should practice good computer security. Finally, shortly before leaving that job, I saw signs that people were listening and responding. I overheard executive-level and senior managers discussing the importance of installing anti-virus software, ensuring software copyright compliance, and accrediting systems. They were speaking in a manner which indicated they thought it to be their responsibility to ensure these tasks were accomplished.

Initially, I thought these events to be strange; these people were talking about my responsibilities as if they were their responsibilities. It finally struck me that I was seeing some positive results of my four-year-long attempt to get people to integrate some of the computer security tasks which I had been preaching and writing about into their regular day's work.

A Commonly-Held View of Computer Security

For years my job was thought of by others to be an unwelcome and possibly not even a necessary evil. It was easy to see how people perceived computer security in such a negative manner. Many computer security programs are born soon after an inspector general's (IG's) visit during which an agency is cited for not having a required program. Some agencies begin a computer security program following a major security incident, or just prior to an IG visit, hoping to avoid yet another adverse finding. In many cases, the embryonic computer security program is placed under an established program (e.g., IRM or physical security) for protection and nurturing while the program matures. Other times, the new program is "thrown to the wolves" - placed in the agency where it must fight for credibility and survival from the very beginning. Regardless of where the new security program is placed, the general perception throughout the agency is that this new program:

- * is the result of a kneejerk management reaction;
- * is just another overhead function (e.g., costs too much and takes away valuable resources that could be better used to do the real job of an agency);
- * does not help other traditional agency functions do their job; and, therefore
- * is not necessary and should not be taken seriously.

This is what I faced when I began my computer security career.

These perceptions were reinforced on a regular basis by data processing managers, computer specialists in the data center, systems planning specialists, functional managers (data owners), and end users. On one occasion I was in a system implementation planning meeting with managers and analysts from data processing, systems planning, contracting, training, and the end user's shop. The local project manager was on the speaker phone with the function manager at our main office, when the function manager mentioned there was a "computer security troublemaker down there" who had raised a stink over what we in the computer security office had seen as a potentially troublesome aspect of connectivity within the proposed new system. I was that troublemaker.

Do Not Allow Others to Shape Your View of Computer Security

For some time, partly because of the initial responses to the new computer security program, I allowed myself to believe that the job of "doing" computer security was somehow outside the mainstream of "doing" agency business. It was only through hard-headed determination that I was able to get invited (or invite myself) to senior management meetings, to meet with project managers, and convince some managers and users, though certainly not all, that they needed to "do" computer security.

An Approach That Works

I began to change the tack I took when "selling" computer security. I realized there are some parts of the job that only the computer security program officer/ manager can do, for example, developing the section of the annual report which shows the status of the security program, maintaining the system accreditation and certification program, providing a head count of personnel with collateral-duty assignments in helping to accomplish computer security tasks, and providing advice and knowledge of available controls and security tools. However, the real work in the computer security program were the day-to-day details, and that was really someone else's job. It was the functional managers', supervisors', system and data owners', and users' job to do computer security. All I had to do was show them the following:

- * what has to be done;
- * why it is their job to do it;
- * why it is in their best interest to do it; and finally,
- * how to do it.

In addition to serving as the agency's Computer Security Officer, I managed a small data processing/information technology (IT) shop. We supported users of microcomputers, LANs, and minicomputers. I managed computer security and the computer support functions, including life cycle management (LCM) and the computer-related portions of the agency's annual business plan and IT budget. With a relatively small staff, including some people physically located in and working for other departments, the challenge quickly became

finding a way to get all of the important work completed. Microcomputer and LAN support (customer support) was a top priority. But so was LCM, since without prior documentation and LCM approval for a new system or systems, funding would not be provided by the ADP/IT budget shop at our headquarters office. Getting some very visible and required computer security tasks completed (e.g., password and access control hardware and software installed on systems, installing anti-virus and access warning message software) was also necessary. It is always useful to have concrete accomplishments early in a program.

When I arrived at that job in 1988, computer specialists and assistants were providing customer support. No one was managing or doing computer security, LCM, or IT-related budgeting. I had heard horror stories in the past about computer security officers spending all of their time "doing" computer security - changing cypher locks, mailing out passwords in envelopes every quarter, and installing password/access control and anti-virus software. These tasks should have been accomplished by computer support staff and/or people who had collateral-duty computer security responsibilities. In my first meetings with executive level and senior managers, I explained that my plan for computer security included those tasks I would do and those tasks staff and computer security collateral-duty personnel would do.

You Must First Get Executive-Level Buy-In

The first "sell" was that I could not do it all, that I had been hired to manage the program, and that there was a long list of requirements - too long a list for one person to accomplish. It was probably to my benefit that I was hired four months before a visit by the inspector general (IG). My audience knew there was a lot of work to do; they had hired me at the last minute to bail them out. Virtually no work had been done in the three years since the last inspection. Selling senior management on the concept of how I would get the work done was not as difficult in this case. In this instance, the agency - specifically, the Commanding Officer and Executive Officer - had hired me as the expert. They expected and trusted me to know how to get the work done. Conversely, if my position had been filled from within the agency, that person would have to gain experience and knowledge before they would have gained credibility. Until then, every recommendation and action may have had to be justified many times over.

Executives and senior managers are more likely to be responsive to the need for computer security since they can see the harm a breach of security can do to an organization. Functional managers are more focused on resource issues, and will usually need to be sold on the idea of improving their program and not putting their resources at risk. However, after senior management buy-in, functional managers' requests for computer security-related resources may be better received and funded.

Selling what had to be done also went well. I did not glaze their eyes over by rattling off all the well known computer security terms and tasks - security plans, accreditation/certification, identification and authentication, auditability, risk analysis, contingency planning, etc. - the all-too-often poorly explained concepts, requirements, and related paperwork that give many managers the perception that we may, indeed, be un-necessary evils. I donned the "plaid used car salesperson's jacket" and told them what they wanted to hear, and then what they

needed to hear. What they wanted to hear was that I had a plan for starting the computer security program, and that the plan could be put in place before the IG visit.

What they needed to hear took a little longer. I began by defining a very important phrase - "computer security". I told them the program should be called "data integrity" or "system integrity," leaving out the word "security." I mentioned that I did not approach the job from a "locks, bars, and guard dogs" perspective, that is, from a traditional physical security program perspective. I did not say this to discount the importance of the physical security aspects of the computer security program, but rather to break down the often pre-conceived negative notion of any program that contains the word "security."

Shortly after I arrived we began to purchase a significant number of microcomputers for the office. The agency's Commanding Officer was already concerned with the increasing value of the hardware and software for which he was responsible. Physical security of these resources was an easy sell. He asked me about a PC-based access control system to control physical access to the offices. He had identified and accepted his responsibility for this aspect of security. Naturally, I agreed with his solution. It was well thought out and appeared to mitigate the threat posed by a less-than-adequate door lock. As valuable as a risk analysis can be to identify the proper control for a particular threat, in some cases (e.g., when the agency director buys-in to your computer security program, has a viable solution to a problem, has the funding, and is volunteering to fund the project) it is prudent to implement it.

Building Credibility Builds a Credible Security Program

A key task and perhaps the most important part of managing a computer security program is building your own credibility. You must make sense to executives and senior management in order for middle management and supervisors to take you and your program seriously. You have to make sense to all these managers before users and operators will integrate security into how they accomplish their jobs. As in the example I have used throughout this paper, the first meetings with executives and senior management are the most important. These first impressions will probably last as long as you are in your position, or as long as these managers are with your agency. A good start can pave the long road you face and make the struggle easier. Conversely, a bad beginning can almost guarantee a difficult, if not impossible, future for you and your security program.

Make Computer Security the Managers' & Users' Job

Some of our systems processed and stored information subject to the Privacy Act of 1974, as well as sensitive management, financial/budget, proprietary, and privileged information. I explained that management had a legal responsibility to adequately protect sensitive information, and then mentioned the Privacy Act and Computer Security Act requirements and penalties. In addition to the data sensitivity issues, I mentioned that the information in the systems was their information in their systems. I mentioned that if their information was important enough or voluminous enough to justify the purchase of a microcomputer to store, process and manage the data, and that if the information were not available to them when

they needed it, or the printed output was not what they had expected, that the accomplishment of their mission or the agency's mission might be adversely affected.

I planted ideas about knowing and controlling who was using their systems, whether their data was being backed up, and how their important work would be accomplished if the systems could not be used. I followed up with conversations about protecting their systems with password and access control hardware and software, backups, and contingency planning, and tied those requirements to the previous conversation. Following on the heels of why it made sense (or should make sense) to managers to protect systems, introducing these concepts was reasonably well received. I wanted to get them thinking about their data as being very closely related to their job. I wanted to persuade them to begin taking their data and its security seriously, and that computer security is not something new to them, it is just another aspect of responsible management. In time, I was able to convince most managers.

Select Your Battles - Pick the Fights You Can Win

I mentioned the software copyright license issue. This was another real-world issue management could understand. They had a legal responsibility to ensure that people in their charge understood and complied with software copyright licenses for the software they used. I explained it, not as something new to worry about, but as something we should have managed all along. I drove the point home by giving examples of cases in which vendors' lawyers sat across the table from government lawyers discussing violations of copyright licenses and the penalties that could be levied. During the first discussion on this subject, senior management dictated on-the-spot that we would make sure we were legal and stayed legal. This up-front understanding and support by senior management paid off later. During follow-up meetings with functional managers I explained the need to upgrade off-the-shelf software, or buy additional copies for new users. I went into such meetings "dual-hatted" - computer security officer and data processing director. Rarely did I have to re-visit my "sales pitch" on the need to comply with copyright licenses. The biggest battle was whether the functional manager would use his or her funds, or whether I would purchase the software from the computer operations and security budget. In cases where the funds were not available we would approach senior management. The discussion and decision never included circumventing the terms of the copyright license. Granted, it was easier in this case being dual-hatted, but when presented with a clear picture of what must be done, and with senior management's support, most managers, even data processing directors, can make the computer security job easier.

We discussed acquiring battery backup devices for minicomputers, LAN servers, and critical microcomputers in the office. We also determined that purchasing a surge suppressor with each new system or purchasing one for each existing system was such an easy way to protect systems and data. The Commanding Officer and Executive Officer mandated that each system would have a surge suppressor. Requests for the necessary funding for battery backup systems and surge suppressors, as well as other computer security resources (e.g., anti-virus software, PC-based access control software and hardware, and LAN-based

software monitoring and audit trail software), became part of the activity's business plan and became items that the Executive-level managers fought for during annual budget negotiations.

Later, as computer viruses became more prevalent, and television and newspapers covered this new threat, I began briefing top management and functional managers, as well as users, about viruses and about how they could prevent virus attacks. With all the media coverage and some successful attacks on some of our systems, the job of convincing top management of the need for local policy and procedures, training, and disciplinary action (when deserved) was an easy task. Management clearly understood the impact on their job or mission which had resulted or could result from the loss of data, processing time and labor. Often my pitch during the Commanding Officer's morning meeting was preempted by my boss advising his department directors and other managers of the latest virus attack or the latest discovery of an infected diskette, before it became an attack. He presented the same news I had passed to him that day or the previous day. He took the matter quite seriously; he viewed computer security vulnerabilities as another threat to doing business. He took it as his job.

Managers clearly understood that when their boss declared a new security policy or verbally reinforced his existing policy and procedures, it was in their best interest to follow his example. Hearing a warning from the Commanding Officer carried that extra amount of clout, beyond hearing it from the computer security officer. This worked especially well in those few cases in which I could not convince the functional manager of the importance of following agency policy. During five quarterly "Captain's Calls" (agency-wide, "all hands" meetings) in the last two years of my tour at that agency, Commanding Officers presented awards and certificates to computer specialists and users who prevented virus attacks by scanning diskettes, promptly reporting the discovery of viruses, and performed other noteworthy security and computer operations tasks. One award was monetary; the others were individualized and highly-coveted coffee mugs. During those years, my staff and collateral-duty security appointees vied for this form of recognition. Their supervisors and managers shared the spotlight when an employee was selected by senior management for an award. By successfully selling the need for vigilance, along with developing reasonable policy and procedures, and training managers and users, the anti-virus portion of the computer security program, for example, took on a life of its own. It literally ran - or could have run - without me.

It is important that users believe they will be rewarded, not punished, for bringing attention to computer security problems. A reward system makes people want to report incidents or unusual events, instead of trying to hide problems or ignore potential problems. Rewarding people for computer security awareness has the added benefit of making management aware that threats do not disappear even if you have a computer security program in place.

Citing aspects of the computer security program that managers could understand got their attention and their acceptance, for the most part. I had used the "appeal to reason" approach - how their jobs and blood pressure could be affected by a preventable system failure, data integrity problems, or the inappropriate disclosure of certain information. I then mentioned in an "oh, by the way" manner, references to federal law. Just telling managers they had to do something because of an instruction from higher authority had never "sold the car" in the

past. Adding the eye-glazing computer security buzzwords and phrases never held their attention, either. As I entered a discussion with managers and users whose expertise was in subjects other than computer operations or security, I reminded myself that I had to translate our profession's jargon into terms that the listener could grasp. Although this is generally thought to be nothing more than a good communication skill, it can make the difference between being understood, accepted, and successful, or being misunderstood, ignored, and perceived to be a failure. The direction of an agency's computer security program is often directly related to the agency's perception and treatment of the security program manager.

Get the Right People to do Computer Security

In order that I could concentrate on "managing" computer security and other functions, I added microcomputer, LAN, and minicomputer security tasks to my computer support staff's daily and weekly list of projects. By making computer security tasks just another part of the job, the following three audiences saw the integration of security disciplines which many people had perceived to be separate functions:

- * The computer specialists and assistants began to view the security work as just another item on the continuous list of things to do;
- * System users saw the computer support people dealing with security issues; and
- * Managers and supervisors saw more people than I doing computer security.

This reinforced the idea that computer security is not outside the mainstream of daily computing and computer support. Over time, I also added computer security responsibilities to the computer support staff's position descriptions and performance plans. Tasks assigned to computer specialists and assistants included:

- * installing anti-virus software;
- * installing an access warning message;
- * installing password and access control software;
- * conducting inventories of hardware and software;
- * completing risk assessment documentation with user assistance;
- * reviewing LAN, minicomputer, and the office access control system (Physical Access Control Management (PACMAN) System) audit logs; and
- * interviewing users to develop a scenario from which to build a contingency plan.

Discover the Value of Collateral-Duty Security Personnel

In order to spread the security workload throughout an agency, especially a large agency, computer security officers can promote the use of collateral-duty security personnel.

Collateral-duty security personnel are those individuals appointed to provide part-time assistance to the computer security office. Collateral-duty appointees can assist in developing security plans, completing risk analyses, conducting hardware and software inventories, coordinating computer security training for system users, reviewing system audit trails, and reviewing user requests for access to systems.

Use of collateral-duty personnel in agencies allows the computer security officer to more effectively implement and maintain a security program. Collateral-duty personnel can be appointed in the following manner:

- * **Local Area Network (LAN) Security Officers:** Each division, branch, and other organizational element which has a LAN, or is planning for the installation of a LAN, could appoint a LAN Security Officer. A system administrator for a LAN could also serve as the LAN Security Officer. The LAN Security Officer can be responsible to implement procedures designed to control access to the LAN, periodically check the LAN server(s) for viruses, perform regular backups, assist with risk analyses and contingency planning, provide basic security training to new users and periodic updates to regular users, and troubleshoot computer security problems.

- * **Information System Security Officers (ISSOs):** ISSOs are made responsible for ensuring computer security policy and procedures are properly implemented. The system administrator for a minicomputer, server, World Wide Web site, firewall, or an e-mail system can serve as the ISSO for that system. Personnel in a data center who are responsible for receiving user requests for system access, establishing accounts, and maintaining user IDs and passwords could be appointed as ISSOs to handle access control functions. Other individuals in the data center could also be appointed as ISSOs for specific systems and data center operations, such as the tape library. The supervisor or manager of a function that uses a number of microcomputers could be appointed as the ISSO of systems for which they are responsible. Some government agencies use titles of Office Automation Coordinator, Office Automation Security Officer, or Microcomputer Area Security Officer to differentiate between collateral-duty security positions responsible for microcomputers and those responsible for larger systems.

- * **Terminal Area Security Officers (TASOs):** Divisions and other organizational elements with **only** remote terminals could appoint TASOs. The TASO can be responsible to implement procedures designed to control access to the terminal area and to troubleshoot computer security problems. The TASO could also serve as the data owner's or application owner's representative and request user access from the appropriate ISSO.

The title of the collateral-duty person is less important than assigning appropriate tasks to the correct individual. These responsibilities are best assigned to the individual who manages, administers, or otherwise has responsibility for the system. The collateral-duty person may

be, or report to, the data owner, functional manager, or data processing service provider. The collateral-duty appointee and his or her management chain have a vested interest in the integrity and security of the system and its data.

In addition to appointing collateral-duty personnel for systems, networks, and terminals, some agencies have found it beneficial to establish a single point of contact for all computer security program administration at each division or major organization level. Some agencies use the title of Security Coordinator. In addition, agencies may consider establishing collateral-duty titles of Assistant ISSO and Assistant LAN Security Officer for those larger offices and agency divisions which have microcomputers and LANs throughout the organization.

Appointing Security Coordinators will allow the agency's computer security officer to distribute requests for accomplishment of computer security tasks to the directors of that agency's major organizational elements. The managers could pass the requirement to their Security Coordinators. Each Security Coordinator could determine whether the request for assistance was related to LAN, microcomputer, or terminal security. The Security Coordinator could then pass the request (or pass their own request) to the appropriate ISSO or LAN Security Officer, via the appropriate division, branch, or other element manager. It is important to utilize the agency's existing chain of command. In a large division, for example, the ISSO for microcomputers could pass the requirement to Assistant ISSOs who are appointed in each branch. The Assistant ISSOs would then work with supervisors and system users to accomplish the task(s). Security Coordinators also offer a way to pass on timely information on new threats, reminders about good practices, and can act as reviewers of policies and procedures.

Some federal agencies have found this hierarchy makes implementation and maintenance of policy, procedures, and practices more manageable, negates the possible impact of distances between some agency offices, and provides easier individual identification and auditability for the computer security officer. Utilization of this method can help spread the workload more evenly among system users and system administrators. This method can increase the agency-wide awareness of information systems security responsibilities, while utilizing the existing management structure.

Collateral-duty Security Coordinators, ISSOs and Assistant ISSOs, and LAN Security Officers and Assistant LAN Security Officers should be trained by the computer security officer. The computer security officer may also want to do some training by contracting with an outside trainer, or by sending people to courses offered outside the agency. Collateral-duty personnel responsibilities should be documented in the agency's computer security policy document(s).

Collateral-duty personnel can report to the computer security officer on information systems security matters, but should use the existing "chain of command" to return or forward documentation to the computer security office. Likewise, the computer security officer can send requests for information and accomplishment of tasks to the directors of the major organization elements (e.g., divisions, directorates, offices). It is crucial to the success of

the computer security program that managers and supervisors be aware of what is being asked of their collateral-duty personnel, as well as the information being passed back to the computer security officer.

Some agencies have incorporated these collateral-duty responsibilities into position descriptions and performance standards. This helps formalize the collateral-duty appointments. It also helps to motivate and provide a basis to award appointees.

When agencies implement a formal program of appointing people to serve as collateral-duty support for the computer security program office, the program can, at times, appear to be running itself. As the program matures, periodic requests by the computer security officer to collateral-duty appointees for information, completion of forms/surveys, or for the training needs of system users, are met without the initial responses, questions, or quarrels. The collateral-duty people, in time, come to respond like extensions of the computer security office.

Conclusion

All of this is not to say that there is an easy-to-follow recipe for success in building a computer security program. Every good program needs five key elements:

- (1) a stable security program management element;
- (2) the existence of an agreed upon, published mission and functions statement;
- (3) the existence of comprehensive, organization-wide information systems security policies;
- (4) a stable resource base; and
- (5) the involvement of the computer security element in the strategic information technology decision-making process.

The key to achieving these elements is a business-oriented approach to integrating computer security into the organization's business or mission, and a healthy dosage of salemanship to make that approach meaningful to those who can integrate computer security into the organization's decision-making process. Once that is accomplished, spreading the work to the organizational elements through the effective use of computer system and network support people and collateral-duty security personnel will allow the security program to grow and mature, in step with the organization's business.

Secure Internet Commerce -- Design and Implementation of the Security Architecture of Security First Network Bank, FSB.

Nicolas Hammond
NJH Security Consulting, Inc.
211 East Wesley Road
Atlanta, GA 30305-3774
Tel: 404-262-1633
Fax: 404-812-1984
njh@njh.com
<http://www.njh.com>

Abstract

Security First Network Bank (SFNB) (<http://www.sfnb.com>) went on-line in October 1995 as the world's first on-line bank. The paper discusses how the security architecture was designed and implemented using the most currently available security technologies.

The encryption technologies used to transport information across the Internet are widely known. Less widely known is how to protect the systems that are directly connected to the Internet, but must interact with customers and protect sensitive information.

This paper discusses the measures that were taken to ensure that SFNB is as safe as possible against hackers. Not only did the entire system have to be safe against attacks but the systems also needed to have the security and assurance needed to meet the Office of Thrift and Supervision approval.

Design Process

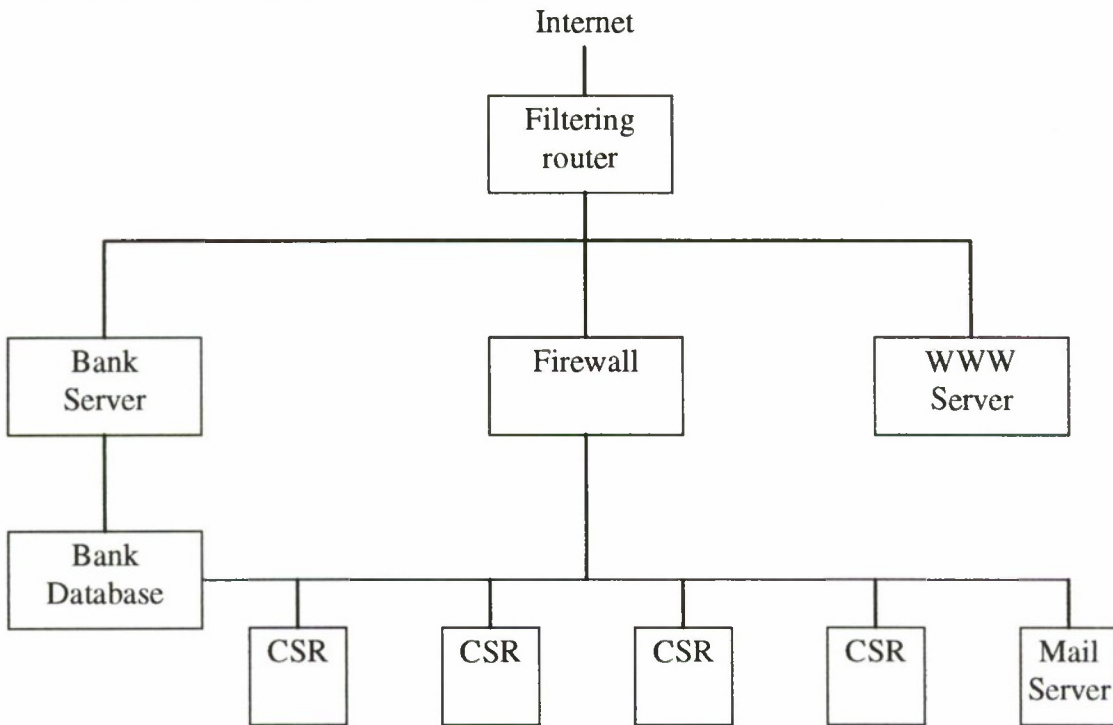
The design of the security architecture began in early 1995. A security architecture paper was written and reviewed by security experts. The design goals were to use sufficient security technology to protect all parts of the bank operations, but at the same time to create a system that could be easily administered by competent system administrators.

The main difference between protecting an on-line bank, and providing a firewall for a commercial company, is that the bank database must be accessible to outside users and therefore has to be protected by a much stronger machine than a conventional firewall. A large part of the architecture discussed how to protect this machine, what type of machine it should be, and the administrative controls that would be needed.

After thorough reviews, the architecture was approved and a security policy was written based upon the security architecture. The time taken to correctly design both the security architecture and the security policy was well spent as it has not been necessary to substantially change either.

Machine Architecture

The security of the entire system rests upon the security of every machine in that system. If a single machine is vulnerable, then potentially the whole network is vulnerable. One of the first stages of the overall design was understanding the data that would flow through the system. Careful consideration was then placed on what machines were needed, and how the machines would be connected.



Except for the Bank Server, this is a fairly typical commercial firewall implementation.

Filtering Router

The filtering router prevents IP-spoofing attacks by ensuring that no incoming packets have an “inside” address. The filtering router also implements filtering rules for each machine to ensure that only authorized traffic is sent to this machine. For example, the WWW server should only receive packets destined for TCP port 80 (http port), the Bank Server should only receive packets for TCP port 443 (https port). The filtering router logs all errors. These error logs have been an invaluable source of information.

WWW Server

The WWW server provides information on the bank to potential customers. The only port that is open on this machine is TCP port 80 (http port). All other network services are de-configured from this machine.

Firewall

The firewall is a dual-homed bastion host running a mail application proxy. Mail is the only traffic allowed to pass through the firewall to the internal network. The firewall also

acts as the Domain Name Server (DNS) for the machines in the DMZ. The firewall selected was *Interceptor*, a firewall product from Technologic (<http://www.tlogic.com>).

Customer Service Representative (CSR) Network

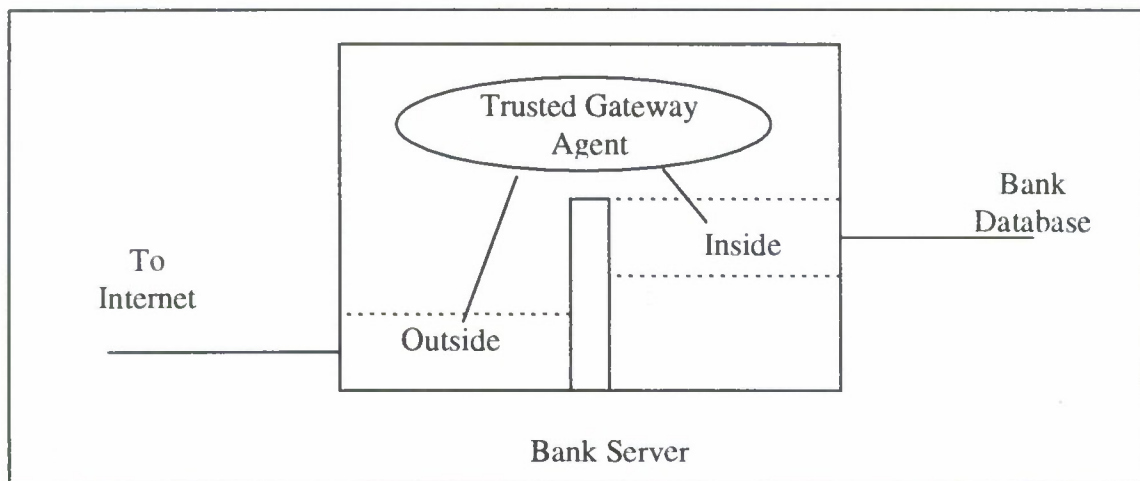
The CSR machines are on a dedicated network. Each CSR has a machine that can be used for receiving and sending mail (stored on a central mail server, external mail sent through the firewall). Each CSR also has read-only access to portions of the bank database to handle bank customer telephone queries. All database actions are audited and can be traced to an individual CSR if necessary.

Bank Database

The Bank Database is the heart of the bank's data center. The machine is dual-homed, with one network card connected to the CSR network to allow CSR read-only access to the database and the other network card connected to the Bank Server. Although all machines need to have a high availability, this machine also needs very high data integrity and so RAID disks are used.

Bank Server

The Bank Server is the machine that protects the bank database. Whereas most firewalls protect internal clients, this machine protects an internal server and therefore needs to be more secure than a firewall and have very strong assurances that it cannot be compromised. Because of these requirements, this machine uses the multilevel secure HP/UX CMW. All features of a trusted system are used -- least privilege, discretionary access control (DAC), mandatory access control (MAC), system integrity and audit.



The Bank Server is dual-homed. One network is connected to the "outside" and runs the Netscape Commerce Server to provide encrypted web traffic through the Secure Socket Layer (SSL). The other network card connects to the "inside" and the Bank Database. Because the "outside" and the "inside" need to be kept totally separate, this machine uses the multilevel capabilities of the HP/UX CMW and assigns a different category for each network. A small trusted program connects the Netscape Commerce Server (running at the "outside" level) with the actual bank applications that run at the "inside" level.

The use of different categories, “inside” and “outside”, provides very strong assurance to the system. Only the Netscape Commerce Server is listening on the “outside” network, all other network services are de-configured. The Netscape Commerce Server is running at the “outside” level. According to the HP/UX CMW rules, a process can only write to files at the same level. The only files at the “outside” level are the Netscape log files.

There are very few applications that make use of the *chroot* capability on Unix systems. One of the design features was to make as much use as possible of the underlying security mechanisms. Therefore, the Netscape Commerce Server runs in a *chroot-ed* environment that only contains the files needed to run it. Note, that the Netscape Commerce Server is started in a *chroot-ed* environment - this is different from using the *chroot* capability of a web server. Therefore, even if there was a bug in the Netscape Commerce Server that allowed an outside user to create a shell, this shell would be running in a *chroot-ed* environment with very few files in it, could only write to the Netscape log files, and is running at an “outside” level so could not even see any files or networks on the “inside.”

A side benefit of this architecture is that it is difficult to mount an internal attack as there is no easy way to export information from the database to the outside world.

System Architecture

Not only must the system be safe against attacks from the Internet, but the overall system must have high availability, strong physical security, data integrity and security from Mother Nature.

High Availability

All systems are connected to Uninterruptible Power Supplies (UPS). Where needed, RAID disks are used.

Physical Security

There is physical security at SFNB’s data center and an even higher level of physical security in the actual machine room.

System Administration

System administration is on a 7x24 basis. All system administration is planned in advanced and with notification to all other system administrators of any scheduled activities. All administration of secure systems is logged and the logs independently checked on a regular basis. This is an expensive procedure, but a necessary one for any secure site.

Training

System administrators receive the necessary training to administer all systems.

Constant Monitoring

Various security related newsgroups and mailing lists are tracked. This is important because if there is a general security alert, SFNB can respond with information posted on its home page. This pro-active security awareness stance helps the company's reputation.

Authentication

There are several authentication problems with all Web based solutions.

The first problem is where should the authentication database be stored. Most Web servers provide an authentication capability implemented by storing username/encrypted passwords in an authentication database readable by the Web server. However this solution opens up security concerns for a secure site.

If the Web server must read the database, then if someone can exploit a hole in the Web server (despite all their claims, most Web servers have had some security problems in their short lifetime), the attacker also can read the authentication database. Admittedly, the database has encrypted passwords, but experience has shown that some passwords can be easily "cracked". If you are a bank, then even the fact that someone has a copy of your authentication database is enough for the public relations department to have the worst nightmares.

Other problems also occur if the authentication database is stored on the "outside". When a new account is opened, the new account name and initial password must be stored in the database. If this database is on the "outside", then there is some administrative interface running on the "outside". This was against the original security architecture which stated that no "trusted" code should be running on the "outside".

SFNB stores the authentication database on the Bank Database machine. This is a protected machine running on an inside network and allows for easier administration and much greater security.

Another authentication problem for Web servers is maintaining an authenticated state. The HyperText Transfer Protocol (HTTP) is supposedly stateless, however it does allow the passing of username/encrypted password as part of the protocol. Once someone has authenticated to the bank, they should only be allowed access to their account. This can be difficult to enforce as a CGI program cannot determine who the authenticator was.

To solve this authentication problem, Netscape "cookies" (see http://www.netscape.com/newsref/std/cookie_spec.html for more information) are used to indicate an authenticated state. Cookies are information that can be passed as part of the HTTP and are converted to environment variables by the Netscape Commerce Server.

New Customers

Much thought was given to the authentication scheme that would be used for customers. Several options were reviewed. Most were too expensive (smart cards) or too impractical (requiring customer to have a separate finger-print/retina reader). The common user/password combination was finally chosen as an interim step to browser client-side authentication.

New customers are assigned a random password. This is sent via regular mail after the customer's address has been verified through a credit-checking agency. When this password is first used, the customer is forced to change the password and chooses their own password. Strict checks are made that the password is not a simple or easily-guessable password.

Maintaining Authenticated State

Once someone has authenticated, there is the problem of connecting the authenticator (the one who supplied the password) to the account he or she is authorized to access. This is solved by using the "cookies" described above.

Because the bank Common Gateway Interface (CGI) programs were the ones that generated the original "cookie", the same programs can verify that the "cookie" belongs to the authorized bank customer. The bank maintains a database mapping a "cookie" to an account. Additional checks are made that the "cookie" expires and also is regenerated each time (to prevent replay attacks).

Encryption

The bank uses the Netscape Commerce Server's SSL to provide encryption services.

Security Technology Employed

The overall architecture employs most of the security features found in a B1/CMW system.

Least Privilege

Least privilege is used in the trusted program that communicates between the "outside" and "inside". The amount of trusted code in this program is very small and can be (has been) examined by inspection. This code is at the heart of the security of the system and is small enough, with enough documentation, to easily pass an A1 evaluation.

Mandatory Access Control (MAC)

MAC is used to provide the information separation to keep the "outside" and "inside" levels apart.

Discretionary Access Control (DAC)

The Netscape Commerce Server runs under a pseudo-user identity. This pseudo-user has read access to the files needed to run the Netscape Commerce Server and has write access to the Netscape log files. The pseudo-user has no other DAC rights on the system.

The Netscape Commerce Server is run in a *chroot*-ed environment providing further protection.

Audit

The system audits all appropriate information. The audit logs are reviewed on a frequent basis to determine any problems with the system.

System Integrity

The Bank Server uses system integrity to determine if any of the system files changed. System files can change either through system administration, or, in a worst case as part of a disk failure. The system integrity tools are run periodically by the Security Officer.

Potential Attacks

The following is a list of possible attacks, and how the system is designed to prevent the attack. SFNB does not divulge the type of attacks that may have been attempted.

IP Spoofing

The filtering router prevents IP spoofing.

User Name Spoofing

The authentication system prevents someone from pretending to be another user by requiring passwords to access the bank, transmitting all passwords encrypted, and using encrypted one-time "cookies" to maintain the authenticated state.

Attempts to Crack Authentication Database

Customer account information is stored on the database server which is protected behind a firewall and the HP/UX CMW. The database cannot be downloaded from the Internet.

Web Server Based Attacks

Attacks against the Netscape Commerce Server are thwarted because of the *chroot*-ed environment and because the "outside" processes cannot see anything on the "inside". The firewall only allows mail to pass through and uses an SMTP filter. Each machine is minimally configured to only do its job, and nothing more.

Summary

The paper has presented the security architecture of SFNB, the world's first on-line bank. The same architecture can be used for other financial institutions, or any company requiring a high level of security. The paper has also discussed some of the authentication issues associated with creating a secure Web server.

Since this work was done, Hewlett Packard acquired the secure web server technology and other assets from SecureWare and are marketing the secure web server technology as

“Virtual Vault”. The author left SecureWare to form his own security consulting company. SFNB went public in May 1996 and are buying the remainder of SecureWare.

Automatic Formal Analyses of Cryptographic Protocols

Stephen H. Brackin *
Arca Systems, Inc.
ESC/AXS
Hanscom AFB, MA 01731-2116

Abstract

Cryptographic protocols are short sequences of message exchanges intended to establish secure communication over insecure networks; whether they actually do so is a notoriously subtle question. This paper describes results produced by a software tool for automatically proving desired properties of protocols using an extension of the Gong, Needham, Yahalom (GNY) belief logic, if possible, and showing exactly what goes wrong otherwise. The paper gives analyses of three complicated SPX protocols, analyses that reveal serious vulnerabilities. Keywords: Protocols; Authentication; Automatic Analysis; Formal Methods.

1. Introduction

Cryptographic protocols are short sequences of message exchanges intended to establish secure communication over insecure networks. Some do so. Others, including ones recommended by experts, can be subverted by attacks involving modified, replayed, or mislabeled messages [5]. The basic issues are *authentication* (whether participants know who they are communicating with), and *nondisclosure* (whether information is revealed to those not meant to receive it).

There are two main approaches to preventing protocol failure: attempting to construct possible attacks, using algebraic properties of the algorithms in the protocols; and attempting to construct inferences, using specialized logics based on a notion of "belief", that protocol participants can confidently reach desired conclusions.

*The author wishes to thank Shiu-Kai Chin, Grace Hammonds, Randy Lichota, and Jack Wool for their assistance. This work was supported by Rome Laboratory and Air Force Materiel Command's Electronic Systems Center/Software Center (ESC/AXS), Hanscom AFB, through the Portable, Reusable, Integrated Software Modules (PRISM) program, contracts F19628-92-C-0006 and F19628-92-C-0008.

Attack-construction tools include Millen's Interrogator [13, 15, 14] and Meadows' NRL Protocol Analyzer [10, 11, 12]. Inference-construction approaches include the belief logics developed by Abadi and Tuttle (AT)[1], by Gong, Needham, and Yahalom (GNY) [7, 6], and by Syverson and van Oorschot (SvO) [17].

Attack-construction tools address both authentication and nondisclosure, but suffer from a combinatorial explosion in the number of cases they must consider. Belief-logic tools address only authentication, but do not face a combinatorial explosion, are potentially much faster, and are potentially capable of analyzing large, complicated protocols that the attack-construction tools are incapable of analyzing in a reasonable time.

The tool whose results are presented here, the Automatic Authentication Protocol Analyzer (AAPA) uses the belief-logic approach. It automatically proves theorems, about a Higher Order Logic (HOL) formalization of a belief logic extending the GNY logic, using HOL proof tools [8]. For the HOL proof tools, whether a claim is a theorem is determined by *type checking* in a Standard ML (SML) [16] compiler. The correctness of the theorems proved by the AAPA thus *does not depend* on the correctness of the AAPA; it depends only on the correctness of the HOL tools and the SML compiler, which have been used and analyzed extensively.

The AAPA automatically translates between HOL and a simple Interface Specification Language (ISL) [4] for describing protocols; the user need only know ISL, not HOL.

The belief logic used by the AAPA grew out of the GNY logic, as adapted by Gong to eliminate impossible protocols [6], but it extends the GNY logic in several ways. These extensions include the following:

- having explicit pairing and conjunction operators;
- describing protocol properties at intermediate protocol stages;

- modeling protocols that use Message Authentication Codes (MACs), i.e., key-dependent hash functions;
- modeling protocols that use key-exchange functions to generate shared secret keys;
- modeling protocols that use hash codes or other computed values as keys;
- modeling protocols that use multiple public-key or symmetric-key encryption functions, multiple hash functions, and multiple key-exchange algorithms.

Several of these extensions are necessary for analyzing the protocols described in this paper, since these protocols use hash codes as keys and make essential use of taking different hashes of the same password. The AAPA belief logic is described in [3].

It is also worth emphasizing that the original GNY logic existed only on paper; proofs in it were constructed and checked by hand. The AAPA not only gives a machine implementation of a logic, with highly reliable machine checking of the accuracy of proofs, but also constructs these proofs automatically. The AAPA produced the results given in this paper in a matter of minutes; doing much less, constructing machine-checked proofs, by hand, took months [9].

The rest of this paper analyzes three SPX protocols [18] using the AAPA. The paper only gives the basic information needed to understand these analyses; see [3, 2, 4] for complete descriptions of the AAPA's underlying HOL theory, its proof process, and the language ISL.

2. SPX Credentials Initialization

An ISL specification for the Credentials Initialization SPX protocol, taken almost verbatim from [18], follows.

The process created by a new user logging in (*C*) contacts the Login Enrollment Agent Facility (*Leaf*), and sends it *C*'s name, a timestamp, a random nonce, and a hash of *C*'s password, all encrypted with *Leaf*'s public key. *Leaf* contacts a Certificate Distribution Center (*Cdc1*), which sends *Leaf* *C*'s long-term private key encrypted with a *different* hash of *C*'s password, the hash of *C*'s password that *Leaf* should have received, and *C*'s user-ID, all encrypted with *Leaf*'s public key. *Leaf* then sends *C* *C*'s user-ID and *C*'s long-term private key encrypted with the different hash of *C*'s password, all encrypted with the random nonce *C* just provided to *Leaf*. From this, *C* is able to determine its own

long-term private and public keys. It then contacts *Cdc1* directly and obtains a certificate for the public key for its Certifying Authority *Ca1*, with *C*'s name, *Ca1*'s name, and the validity interval for this public key, all signed by taking the data's hash with yet another hash algorithm and encrypting the result with *C*'s long-term private key.

The protocol's ISL specification follows standard notation, uses intuitive terminology, and is largely self-explanatory. The following descriptions of ISL constructs will suffice for getting a reasonable understanding of the protocol:

- The **From** construct on initial hash codes and encrypted values allows the AAPA to compute a putative source for each such value in the protocol; it uses this to direct its proof process.
- $\{x\}f(k)$ denotes *x* encrypted using function *f* with key *k*.
- $[x](f1,f2)(k)$ denotes *x* together with a signature produced by taking the hash of *x* using *f1* and encrypting the result using *f2* with key *k*.
- The **||** operator binds a statement to a data item, as in the "extension" concept in the GNY logic [7]. The protocol assumes that the principal originating this data item will not send it unless this principal is confident that the statement is true.
- ISL accepts and ignores C-style comments.

The protocol's ISL specification follows:

DEFINITIONS:

PRINCIPALS: *C*, *Ca1*, *Cdc1*, *Leaf*;
PUBLIC KEYS: *KpC*, *KpCa1*, *KpLeaf*;
PRIVATE KEYS: *KsC*, *KsCa1*, *KsLeaf*;
SYMMETRIC KEYS: *Rn*;
OTHER: *PwdC*, *UidC*, *ValidityKpCa1*, *Ts*;
ENCRYPT FUNCTIONS: *Des*, *Rsa*;
HASH FUNCTIONS: *H1*, *H2*, *H3*;
Des WITH ANYKEY HASINVERSE *Des* WITH ANYKEY;
Rsa WITH *KpLeaf* HASINVERSE *Rsa* WITH *KsLeaf*;
Rsa WITH *KsC* HASINVERSE *Rsa* WITH *KpC*;

INITIALCONDITIONS:

C Received
Des, *H1*, *H2*, *H3*, *Rsa*, *C*, *PwdC*, *Ts*, *KpLeaf*, *Rn*;
C Believes
(*PublicKey Leaf Rsa KpLeaf*;
SharedSecret *C C H2(PwdC) From C*);
Cdc1 Received
Rsa, *UidC*, *KpLeaf*,
H1(PwdC) From C,

```

{KsC}Des(H2(PwdC))          /* 1 */
  ||(PrivateKey C Rsa KsC) From C,
[C,Ca1,ValidityKpCa1,KpCa1](H3,Rsa)(KsC)
  ||(PublicKey Ca1 Rsa KpCa1) From C;
Leaf Received Des,Rsa,KsLeaf;

```

PROTOCOL:

1. C -> Leaf: {C,Ts,Rn,H1(PwdC)}Rsa(KpLeaf);
2. Leaf -> Cdc1: C;
3. Cdc1 -> Leaf:


```

      {{KsC}Des(H2(PwdC))          /* 2 */
        ||(PrivateKey C Rsa KsC),
        H1(PwdC),
        UidC}Rsa(KpLeaf);

```
4. Leaf -> C:


```

      {UidC,
      {KsC}Des(H2(PwdC))          /* 3 */
        ||(PrivateKey C Rsa KsC)}Des(Rn);

```
5. C -> Cdc1: C;
6. Cdc1 -> C:


```

      [C,Ca1,ValidityKpCa1,KpCa1](H3,Rsa)(KsC)
      ||(PublicKey Ca1 Rsa KpCa1);

```

GOALS:

4. C Possesses KpC,KsC;


```

      C Believes
      (PublicKey C Rsa KpC;
      PrivateKey C Rsa KsC);

```
6. C Possesses ValidityKpCa1,KpCa1;


```

      C Believes PublicKey Ca1 Rsa KpCa1;

```

The remainder of this paper will assume that this ISL specification is in a file named `spxinit.isl`. Running the AAPA on this file gives the error message:

User-goal failure, stage: 4!

Goal statement: C Possesses KpC,KsC;

and produces files `spxinit.fail` and `spxinit.prvd` containing ISL descriptions of the failed default goals and proved theorems. One of the theorems is `C Received KsC`; so the problem is with `KpC`; the proof rules embodied in the AAPA cannot prove that a public key can be computed from the corresponding private key. The assumption that this can be done is implicit in [18]. The necessary machinery to allow the user to specify that a public key is a function of the corresponding secret key is only partially present in the AAPA.

This problem can be easily worked around by replacing `KsC` by `KpC,KsC` and

```
PrivateKey C Rsa KsC
```

by

```
PrivateKey C Rsa KsC; PublicKey C Rsa KpC
```

in the lines marked `/* 1 */`, `/* 2 */`, and `/* 3 */`.

This change causes the AAPA to give the error;

User-goal failure, stage: 4!

Goal statement:

```

C Believes
(PublicKey C Rsa KpC;PrivateKey C Rsa KsC);

```

The `spxinit.fail` file now shows the failed default goal

```

C Believes
(C Conveyed
{KpC,KsC}Des(H2(PwdC))
  ||(PublicKey C Rsa KpC;
  PrivateKey C Rsa KsC));

```

The subgoal the prover is unable to prove is:

```
C Believes (C Recognizes KpC,KsC);
```

The goal asserts that C can be confident that the keys (key, actually) encrypted with a hash of C's password really originated with C. The problem is that any random value can be decrypted with a hash of C's password to produce another random value; how is C to know whether the result is C's secret key, which looks random itself? The failed subgoal says C can identify the decrypted data as meaningful.

This might be a problem. In the real protocol, the data represented abstractly here as `KsC` might have some structure, or be encrypted with identifying information, so that a decrypted random value can be recognized as meaningless, but the analysis here raises the question for implementations of the protocol as to whether they make such tests.

This new problem can be solved by putting identifying information, C's name, in with the encrypted key(s), and adding the initial condition

```
C Believes C Recognizes C;
```

i.e., C can identify its own name as meaningful.

This change causes the AAPA to display the same error message that it displayed before, but `spxinit.fail` shows differences; the top failed default goal is now:

```

[C Believes
(PublicKey C Rsa KpC;
 PrivateKey C Rsa KsC;
 C Possesses Des,H2(PwdC);
 C Possesses C,KpC,KsC);
C Believes
(C Believes
(PublicKey C Rsa KpC;

```



```

PrivateKey C Rsa KsC;
C Possesses Des,H2(PwdC);
C Possesses C,KpC,KsC))]

```

and its waiting subgoals are

```

C Believes
(Fresh C,KpC,{KsC}Des(H2(PwdC))
||(PublicKey C Rsa KpC;
  PrivateKey C Rsa KsC));
C Believes (Trustworthy C);

```

The two parts of this goal reflect that if the belief logic allows the recipient of a data item to believe the properties that the protocol assumes this data item has, then it also allows this recipient to believe that the originator of this data item also believes these properties. The same hypotheses give both conclusions.

The second of the waiting subgoals, that **C** considers itself trustworthy, is trivial and easy to add as an initial condition. The first subgoal, though, reflects a real limitation in the GNY logic.

Following the GNY logic, the **AAPA**'s belief logic does not allow a protocol participant to believe a statement that the protocol assumes is valid for a data item unless this participant has adequate reason to believe that this data item was created for the current protocol run; otherwise it could be a replay. In the current case, **C**'s encrypted private key was not created for the current protocol run, but it has the properties that the protocol assumes it has. As long as the current **PwdC** is no older than the current **KsC**, there is no way for a replay to give **C** a stale private key. The theory's **Fresh** construct, meaning "created for the current run", needs to be generalized to "fresh enough", meaning "having the expected properties for the current run".

This last problem can be solved by having **C** believe that **PwdC** was created for the current run. It might have been, after all, and this assumption accurately reflects that the critical issue is whether **C**'s password is too old.

Adding the initial conditions

```

C Believes (Fresh PwdC; Trustworthy C)

```

gets the **AAPA** past all the default and user-set goals for stage 4. Now it encounters a problem similar to an earlier one:

User-goal failure, stage: 6!

Goal statement:

```

C Believes (PublicKey Ca1 Rsa KpCa1);

```

Again, the problem is that data that was not created for the current protocol run — **KpCa1** — has to be believed to have the properties the protocol assumes for it. Adding the initial condition

```

C Believes Fresh ValidityKpCa1;

```

causes the **AAPA** to prove the all the user-set goals.

The analysis of the **SPX** Credentials Initialization protocol reveals some deficiencies in the GNY-based formal theory underlying the **AAPA**, identifies unstated assumptions in [18], and identifies a potential problem — not checking whether the decrypted private key can be identified as being information of an expected form — that the protocol's implementations might have.

3. SPX Authentication

Although it is more complicated, and roughly 50% more difficult to specify, the **SPX** Authentication protocol raises only issues similar to those raised by the **SPX** Credentials Initialization protocol.

In this protocol, a claimant (**C**), already possessing its own long-term public and private keys (**KpC**, **KsC**), a pair of shorter-term session public and private keys (**KspC**, **KssC**), and the public key (**KpCa1**) of its Certifying Authority (**Ca1**), and already believing that all these keys are what they are, contacts and verifies its identity to a verifier (**V**) already possessing its own long-term public and private keys (**KpV**, **KsV**) and the public key (**KpCa2**) of its Certifying Authority (**Ca2**), and already believing that these keys are what they are.

C contacts a Certificate Distribution Center (**Cdc1**) to obtain a certificate signed with **Ca1**'s private key giving **V**'s public key. **C** then creates a timestamp (**Ts**) and a random symmetric key (**DesKey**), and sends **V** three things:

- An *authenticator*, consisting of **Ts** and **C**'s channel ID **ChannelIdC**, signed with a DES residue of these values produced with **DesKey**. The DES residue is effectively a key-dependent hash.
- A *ticket*, containing the session public key **KspC**, a validity interval **ValidityKspC** for this key, and **C**'s User-ID **UidC**. The ticket is signed with a hash code produced by **H3** and encrypted with **C**'s long-term secret key **KsC**. The ticket communicates **KspC** and identifies it as being from **C**.
- A *delegator*, consisting of **DesKey** encrypted with **V**'s public key, and signed with a hash code produced by **H3** and encrypted with the session secret key **KssC**. The delegator communicates **DesKey** and indirectly identifies it as being from **C**.

After receiving all these things, **V** contacts a Certificate Distribution Center (**Cdc2**) to obtain a certificate signed with **Ca2**'s private key giving **C**'s public key. **V** then checks all the information from **C**, and sends **C** **Ts**

encrypted with **DesKey** to confirm its receipt of **DesKey** and its intent to use it for subsequent communication.

An ISL specification for this protocol follows, including modifications from its description in [18], similar to those for the Credentials Initialization protocol, needed to have it meet all its user-set goals. These modifications include putting the name **C** in with **DesKey** and believing that validity intervals are fresh.

DEFINITIONS:

PRINCIPALS: **C**, **Ca1**, **Ca2**, **Cdc1**, **Cdc2**, **V**;
PUBLIC KEYS: **KpC**, **KpCa1**, **KpCa2**, **KpV**, **KspC**;
PRIVATE KEYS: **KsC**, **KsCa1**, **KsCa2**, **KsV**, **KssC**;
SYMMETRIC KEYS: **DesKey**, **Rn**;
OTHER: **ChannelIdC**, **UidC**, **ValidityKpC**,
ValidityKpV, **ValidityKspC**, **Ts**;
ENCRYPT FUNCTIONS: **Des**, **Rsa**;
KEYED HASH FUNCTIONS: **Hdes**;
HASH FUNCTIONS: **H3**;
Des WITH ANYKEY HASINVERSE **Des** WITH ANYKEY;
Rsa WITH **KsC** HASINVERSE **Rsa** WITH **KpC**;
Rsa WITH **KsCa1** HASINVERSE **Rsa** WITH **KpCa1**;
Rsa WITH **KsCa2** HASINVERSE **Rsa** WITH **KpCa2**;
Rsa WITH **KpV** HASINVERSE **Rsa** WITH **KsV**;
Rsa WITH **KssC** HASINVERSE **Rsa** WITH **KspC**;

INITIALCONDITIONS:

C Received
Des, **H3**, **Hdes**, **Rsa**,
Ts, **ChannelIdC**, **UidC**, **ValidityKspC**,
C, **V**, **DesKey**, **KpC**, **KsC**, **KpCa1**, **KspC**, **KssC**;
C Believes
(Fresh **Ts**;
Fresh **ValidityKpV**;
Fresh **DesKey**;
Fresh **KspC**;
PublicKey C **Rsa KspC**;
PublicKey Ca1 **Rsa KpCa1**;
PrivateKey C **Rsa KssC**;
C Recognizes **Ca1**;
C Recognizes **Ts**;
SharedSecret C V **DesKey**;
Trustworthy Ca1;
Trustworthy V);
V Received
Des, **H3**, **Hdes**, **Rsa**, **Ts**, **C**, **KpV**, **KsV**, **KpCa2**;
V Believes
(Fresh **Ts**;
Fresh **ValidityKpC**;
Fresh **ValidityKspC**;
PrivateKey V **Rsa KsV**;
PublicKey V **Rsa KpV**;
PublicKey Ca2 **Rsa KpCa2**;
V Recognizes **C**;

V Recognizes **Ca2**;
V Recognizes **ValidityKspC**;
Trustworthy C;
Trustworthy Ca2);

Cdc1 Received

[**Ca1**, **V**, **ValidityKpV**, **KpV**] (**H3**, **Rsa**) (**KsCa1**)
|| (**PublicKey V** **Rsa KpV**) From **Ca1**;

Cdc2 Received

[**Ca2**, **C**, **ValidityKpC**, **KpC**] (**H3**, **Rsa**) (**KsCa2**)
|| (**PublicKey C** **Rsa KpC**) From **Ca2**;

PROTOCOL:

1. **C** → **Cdc1**: **V**;
2. **Cdc1** → **C**:
[**Ca1**, **V**, **ValidityKpV**, **KpV**] (**H3**, **Rsa**) (**KsCa1**)
|| (**PublicKey V** **Rsa KpV**);
3. **C** → **V**:
<**Ts**, **ChannelIdC**> **Hdes** (**DesKey**)
|| (**Fresh DesKey**),
[**ValidityKspC**, **UidC**, **KspC**] (**H3**, **Rsa**) (**KsC**)
|| (**PublicKey C** **Rsa KspC**; **Fresh KspC**),
[**C**, **DesKey**] **Rsa** (**KpV**) (**H3**, **Rsa**) (**KssC**)
|| (**SharedSecret C V** **DesKey**);
4. **V** → **Cdc2**: **C**;
5. **Cdc2** → **V**:
[**Ca2**, **C**, **ValidityKpC**, **KpC**] (**H3**, **Rsa**) (**KsCa2**)
|| (**PublicKey C** **Rsa KpC**);
6. **V** → **C**:
{**Ts**} **Des** (**DesKey**)
|| (**SharedSecret C V** **DesKey**);

GOALS:

2. **C** Possesses **ValidityKpV**, **KpV**;
C Believes **PublicKey V** **Rsa KpV**;
3. **V** Possesses
Ts, **ChannelIdC**, **ValidityKspC**,
UidC, **KspC**, **DesKey**;
5. **V** Possesses **ValidityKpC**, **KpC**;
V Believes
(**PublicKey C** **Rsa KpC**;
PublicKey C **Rsa KspC**;
Fresh KspC; **Fresh DesKey**;
C Possesses **DesKey**;
SharedSecret C V **DesKey**;
C Believes **SharedSecret C V** **DesKey**);
6. **C** Believes
(**V** Possesses **DesKey**;
V Believes **SharedSecret C V** **DesKey**);

Although the initial-conditions modifications are similar to those for the Credentials Initialization protocol, these modifications are much more questionable for the Authentication protocol.

In the Authentication case, **C** and **V** initially believe

ValidityKpV and **ValidityKpC** are fresh, and use these “fresh enough” beliefs to believe that each others’ public keys are what they are. That is not troubling, but **V** also uses the dubious belief that **ValidityKspC** is “fresh enough” to derive the even more dubious belief, initially held by **C**, that **KspC** was created for the current run. It then uses this conclusion to derive that **DesKey** is a shared secret between **C** and **V**, and uses this shared-secret belief to derive that **DesKey** was created by **C** for the current run.

Since SPX session keys are intended to be used for up to 8 hours [18], **KspC** is really not “fresh enough”; an attacker having a ticket and the corresponding **KssC** could use a new **DesKey** and **V**’s widely-available public key to fake an authentication transfer and by doing so impersonate **C**.

4. SPX Delegation

The SPX Delegation protocol is very similar to the SPX Authentication protocol. The only difference is that the delegator:

```
[{DesKey}Rsa(KpV)](H3,Rsa)(KssC)
||(SharedSecret C V DesKey);
```

changes to the new delegator:

```
{DesKey}Rsa(KpV),
{KssC}Des(DesKey)||(PrivateKey C Rsa KssC);
```

Note that, since a signed message is a pair consisting of a message and a signature, the delegators in both protocols are pairs whose first elements are **{DesKey}Rsa(KpV)**. The second element of the pair changes, as does the property associated with this element.

The goals for the Delegation protocol are also very similar to those for the Authentication protocol. The single new goal

```
V Believes PrivateKey C Rsa KssC;
```

replaces the two final stage-5 shared-secret goals in the Authentication protocol.

It turns out, though, that it is impossible to get the proofs of the desired properties to go through, even with dubious “fresh enough” assumptions like those used in the Authentication protocol. Inserting a **C** in with the encrypted key in **{DesKey}Rsa(KpV)** and in

```
{KssC}Des(DesKey)||(PrivateKey C Rsa KssC)
```

helps, so that **V** is able to recognize the encrypted information as meaningful, but there is never a reasonable justification for claiming that

```
V Believes (SharedSecret V C DesKey);
```

Because of this, the default goals

```
V Believes
(C Conveyed
  {C,KssC}Des(DesKey)
  ||(PrivateKey C Rsa KssC));
V Believes
(C Conveyed {C,DesKey}Rsa(KpV));
V Believes
(C Conveyed
  Hdes(DesKey,Ts,ChannelIdC)
  ||(Fresh DesKey));
```

all fail. As described earlier, it is possible for anyone holding a valid ticket from **C** and a corresponding **KssC** to create a **DesKey** and fake a delegation request from **C**. Further, anyone to whom **C** has made a delegation request will have such a ticket and a corresponding **KssC**, and these items will be valid for up to 8 hours. The SPX Delegation protocol does not prevent, say, bankers from obtaining their customers’ medical records.

5. Summary

This paper has described a tool that makes it possible to perform careful formal analyses of authentication properties of cryptographic protocols quickly and easily. For the three SPX protocols described in [18], this tool reveals unstated assumptions and a potential weaknesses in the first protocol, and serious weaknesses in the other two protocols. These weaknesses manifest themselves in one case by requiring dubious initial-conditions assumptions to prove the desired conditions, and in the other case by making it impossible to prove these desired conditions even with the dubious assumptions.

References

- [1] M. Abadi and M. Tuttle. A semantics for a logic of authentication. In *Proceedings of the 10th Symposium on Principles of Distributed Computing*, pages 201–216. ACM, August 1991.
- [2] S. Brackin. Deciding cryptographic protocol adequacy with HOL: The implementation. To Appear in The 1996 International Conference on Theorem Proving in Higher Order Logics, Turku, Finland, August 1996.
- [3] S. Brackin. A HOL extension of GNY for automatically analyzing cryptographic protocols. In *Proceedings of Computer Security Foundations Workshop IX*, County Kerry, Ireland, June 1996. IEEE.

- [4] S. Brackin. An interface specification language for cryptographic protocols and its translation into HOL. Submitted to the New Security Paradigms Workshop, Arrowhead, CA, September 1996.
- [5] D. Denning and G. Sacco. Timestamps in key distribution protocols. *CACM*, 24(8):533–536, August 1981.
- [6] L. Gong. Handling infeasible specifications of cryptographic protocols. In *Proceedings of Computer Security Foundations Workshop IV*, pages 99–102, Franconia NH, June 1991. IEEE.
- [7] L. Gong, R. Needham, and R. Yahalom. Reasoning about belief in cryptographic protocols. In *Proceedings of the Symposium on Security and Privacy*, pages 234–248, Oakland, CA, May 1990. IEEE.
- [8] M. Gordon and T. Melham. *Introduction to HOL: A Theorem Proving Environment for Higher Order Logic*. Cambridge University Press, Cambridge, UK, 1993.
- [9] G. Hammonds, R. Lichota, G. Hird, and J. Wool. Command center security — proving software correct. In *Proceedings the 10th Annual Conference on Computer Assurance*, pages 163–173, Gaithersburg, MD, June 1995. NIST.
- [10] C. Meadows. Using narrowing in the analysis of key management protocols. In *Proceedings of the Symposium on Security and Privacy*, pages 138–147, Oakland, CA, May 1989. IEEE.
- [11] C. Meadows. A system for the specification and analysis of key management protocols. In *Proceedings of the Symposium on Security and Privacy*, pages 182–195, Oakland, CA, May 1991. IEEE.
- [12] C. Meadows. Applying formal methods to the analysis of a key management protocol. *J. Computer Security*, 1(1):5–36, 1992.
- [13] J. Millen. The interrogator: A tool for cryptographic protocol analysis. In *Proceedings of the Symposium on Security and Privacy*, pages 134–141, Oakland, CA, May 1984. IEEE.
- [14] J. Millen. The Interrogator model. In *Proceedings of the Symposium on Security and Privacy*, pages 251–260, Oakland, CA, May 1995. IEEE.
- [15] J. Millen, S. Clark, and S. Freedman. The Interrogator: Protocol security analysis. *IEEE Trans. on Software Engineering*, SE-13(2):274–288, February 1987.
- [16] L. Paulson. *ML for the Working Programmer*. Cambridge University Press, Cambridge, UK, 1993.
- [17] P. Syverson and P. van Oorschot. On unifying some cryptographic protocol logics. In *Proceedings of the Symposium on Security and Privacy*, pages 14–28, Oakland, CA, 1994. IEEE.
- [18] J. Tardo and K. Alagappan. SPX: Global authentication using public key certificates. In *Proceedings of the Symposium on Security and Privacy*, pages 232–244, Oakland, CA, 1991. IEEE.

Automatic Formal Analyses of Cryptographic Protocols

**19th National Information Systems Security
Conference**

October 22-25, 1996

Baltimore Convention Center

Dr. Stephen H. Brackin

Arca Systems, Inc.

303 E. Yates St., Ithaca, NY 14850

607-277-8211 or 607-277-2739

brackin@va.arca.com

**Supported by ESC/AXS through PRISM, and
by Rome Laboratory**

Page - 1



Outline of Talk

- **Problem: protocol failure**
- **Automatic Authentication Protocol Analyzer (AAPA)**
- **Three SPX protocols and results of analyzing them**
- **Conclusions, for SPX and arbitrary protocols**

Page - 2



Cryptographic Protocols

- **Goal: Secure communication over insecure networks**
 - Networks, principals, messages
 - Worst case: enemy controls all communication
 - Nondisclosure and authentication
- **Tools:**
 - Shared or confirmable secrets
 - Encryption
 - Hash functions
 - Timestamps, nonces, signatures, key-exchange functions
- **Distributed algorithms**

Page - 3



PRISM
Custom Command and Control

Failure Example

- **Tatebayeshi-Matsuzaki-Newman protocol**
 - 1. $A \rightarrow S: \{K_a\}_{Rsa(Pk_S)}, A, B$
 - 2. $S \rightarrow B: S, A$
 - 3. $B \rightarrow S: \{K_b\}_{Rsa(Pk_S)}$
 - 4. $S \rightarrow A: \{K_b\}_{Des(K_a)}$
- **AAPA notation, but more-or-less standard**
- **Published (CRYPTO '89), recommended by experts**
- **It's wrong --- and has lots of company**

Page - 4

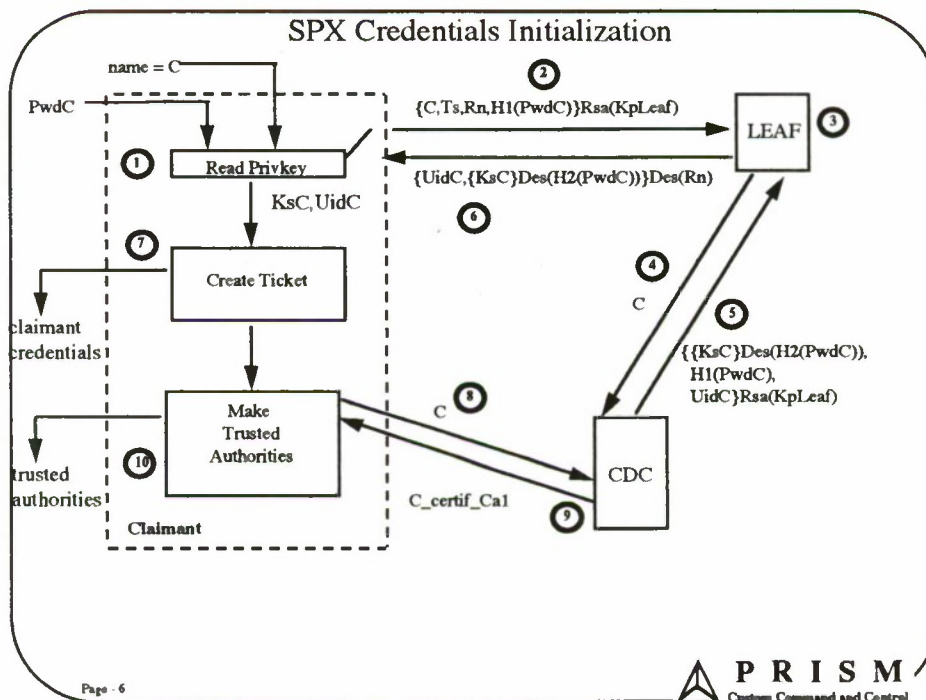


PRISM
Custom Command and Control

Automatic Authentication Protocol Analyzer

- Inputs Interface Specification Language (ISL) specs
- Produces Higher Order Logic (HOL) theories
- Automatically proves default and user-set goals
 - Belief logic extending Gong-Needham-Yahalom logic
 - Sample deduction: If P believes only P and Q know K, and P receives M that K decrypts to something meaningful, then P believes Q sent M --- though not necessarily recently or to P
 - Proceeds by induction on protocol stage
- Gives proof results in ISL

Page - 5

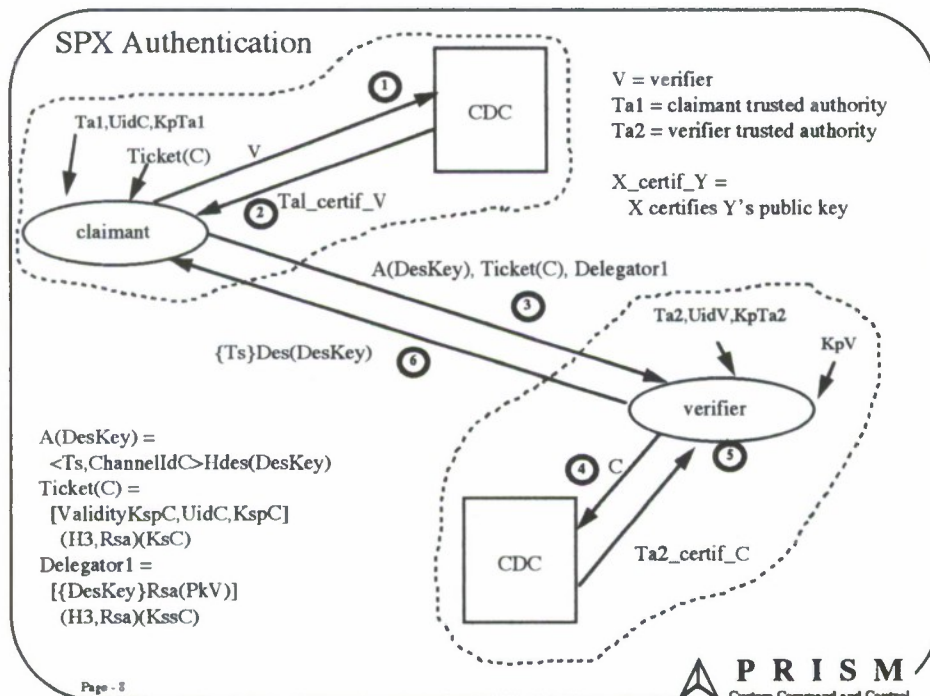


Page - 6

What AAPA Analysis Shows: I

- **KpC must be computable from KsC**
- **Keys must be stored along with recognizable data**
- **PwdC must not be older than KsC**
- **ValidityKpCa1 must include the current time**

Page - 7



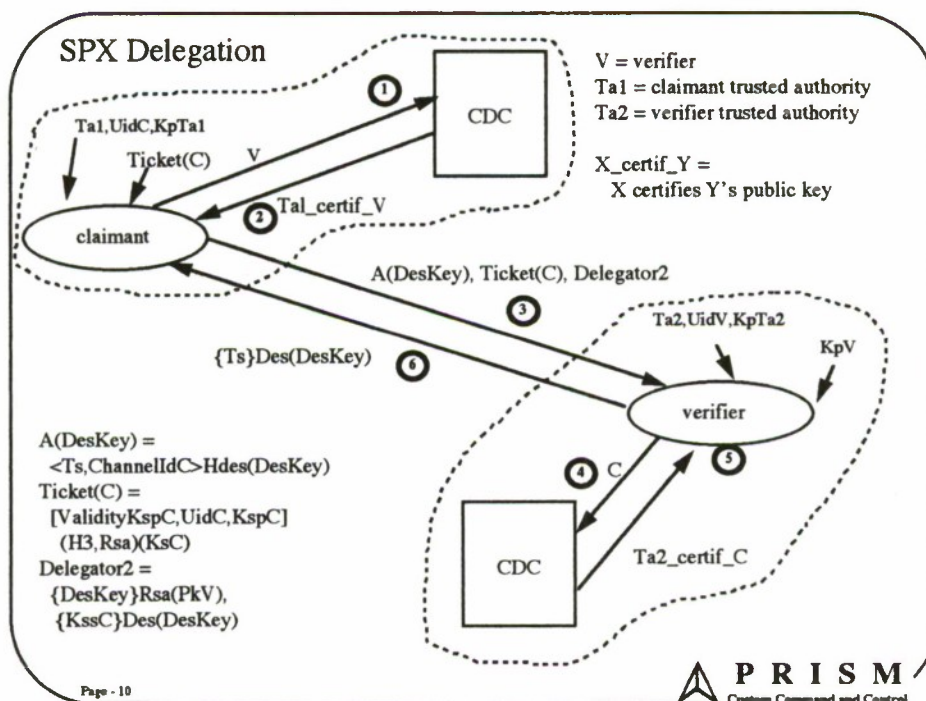
Page - 8



What AAPA Analysis Shows: II

- **Keys must be stored with recognizable data**
- **Validity intervals must include the current time**
 - **ValidityKpV, ValidityKpC, ValidityKspC**
- **Belief DesKey from C depends on dubious assumptions**
- **Delegation gives up to 8 hours of authentication failure**

Page - 9



Page - 10



What AAPA Analysis Shows: III

- **Similar recognizability and interval restrictions**
- **Dubious assumptions don't give belief KssC from C**
- **Banker can obtain medical records**

Page - 11



Conclusions

- **For the SPX protocols:**
 - Initialization must include checks for meaningful data
 - Authentication possibly flawed
 - Delegation possibly flawed
 - These issues should be addressed in documentation
- **For all cryptographic protocols:**
 - The AAPA is a fast, easy tool for reducing failures
 - The AAPA can be used as part of the design process

Page - 12



Surmounting the Effects of Lossy Compression on Steganography

Daniel L. Currie, III

Fleet Information Warfare Center
2555 Amphibious Drive
NAB Little Creek
Norfolk, VA 23521-3225
currie@msn.com

Cynthia E. Irvine

Computer Science Department
Code CS/Ic
Naval Postgraduate School
Monterey, CA 93943-5118
irvine@cs.nps.navy.mil

Abstract

Steganographic techniques can be used to hide data within digital images with little or no visible change in the perceived appearance of the image and can be exploited to export sensitive information. Since images are frequently compressed for storage or transmission, effective steganography must employ coding techniques to counter the errors caused by lossy compression algorithms. The Joint Photographic Expert Group (JPEG) compression algorithm, while producing only a small amount of visual distortion, introduces a relatively large number of errors in the bitmap data. It is shown that, despite errors caused by compression, information can be steganographically encoded into pixel data so that it is recoverable after JPEG processing, though not with perfect accuracy.

1. Introduction

Two techniques are available to those wishing to transmit secrets using unprotected communications media. One is cryptography, where the secret is scrambled and can be reconstituted only by the holder of a key. When cryptography is used, the fact that the secret was transmitted is observable by anyone. The second method is steganography¹. Here the secret is encoded in another message in a manner such that, to the casual observer, it is unseen. Thus, the fact that the secret is being transmitted is also a secret.

Widespread use of digitized information in automated information systems has resulted in a renaissance for steganography. Information which provides the ideal vehicle for steganography is that which is stored with an accuracy far greater than necessary for the data's use and display. Image, Postscript, and audio files are among those that fall into this category, while text, database, and executable code files do not.

It has been demonstrated that a significant amount of information can be concealed in bitmapped image files with little or no visible degradation of the image[4]. This process, called steganography, is accomplished by replacing the least significant bits in the pixel bytes with the data to be hidden. Since the least significant pixel bits contribute very little

1. The term steganography derives from a method of hidden writing discussed by Trimetheus in his three-volume *Steganographia*, published in 1499 [3].

to the overall appearance of the pixel, replacing these bits often has no perceptible effect on the image. To illustrate, consider a 24 bit pixel which uses 8 bits for each of the red, green, and blue color channels. The pixel is capable of representing 2^{24} or 16,777,216 color values. If we use the lower 2 bits of each color channel to hide data (Figure 1), the maximum change in any pixel would be 2^6 or 64 color values; a minute fraction of the whole color space. This small change is invisible to the human eye. To continue the example, an image of 735 by 485 pixels could hold $735 \times 485 \times 6 \text{ bits/pixel} \times 1 \text{ byte}/8 \text{ bits} = 267,356$ bytes of data.

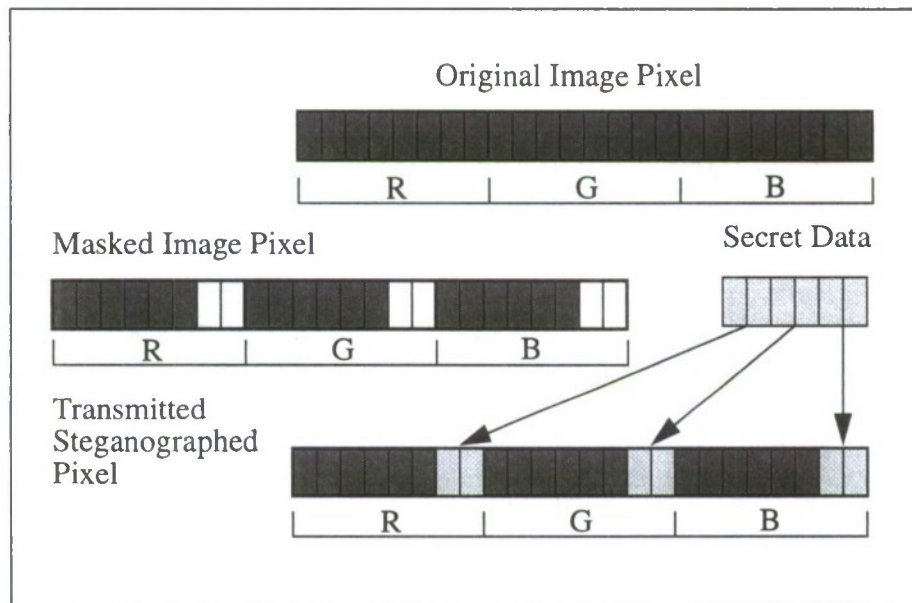


Figure 1:

Kurak and McHugh [4] show that it is even possible to embed one image inside another. Further, they assert that visual inspection of an image prior to its being downgraded is insufficient to prevent unauthorized flow of data from one security level to a lower one.

A number of different formats are widely used to store imagery including BMP, TIFF, GIF, etc. Several of these image file formats "palletize" images by taking advantage of the fact that the color veracity of the image is not significantly degraded to the human observer by drastically reducing the total number of colors available. Instead of over 16 million possible colors, the color range is reduced and stored in a table. Each pixel, instead of containing a precise 24-bit color, stores an 8-bit index into the color table. This reduces the size of the bitmap by 2/3. When the image is processed for display by a viewer such as "xv" [1], the indices stored at the location of each pixel are used to obtain the colors to be displayed from the color table. It has been demonstrated that steganography is ineffective

when images are stored using this compression algorithm[2]. Difficulty in designing a general-purpose steganographic algorithm for palletized images results from the following factors: a change to a “pixel” results in a different index into the color table, which could result in a dramatically different color, changes in the color table can result in easily perceived changes to the image, and color maps vary from image to image with compression choices made as much for aesthetic reasons as for the efficiency of the compression.

Despite the relative ease of employing steganography to covertly transport data in an uncompressed 24-bit image, lossy compression algorithms based on techniques from digital signal processing, which are very commonly employed in image handling systems, pose a severe threat to the embedded data. An excellent example of this is the ubiquitous Joint Photographic Experts Group (JPEG) [7][5] compression algorithm which is the principle compression technique for transmission and storage of images used by government organizations. It does a quite thorough job of destroying data hidden in the least significant bits of pixels. The effects of JPEG on image pixels and coding techniques to counter its corruption of steganographically hidden data are the subjects of this paper.

2. JPEG Compression

JPEG has been developed to provide efficient, flexible compression tools. JPEG has four modes of operation designed to support a variety of continuous-tone image applications. Most applications utilize the Baseline sequential coder/decoder which is very effective and is sufficient for many applications.

JPEG works in several steps. First the image pixels are transformed into a luminance/chrominance color space [6] and then the chrominance component is downsampled to reduce the volume of data. This downsampling is possible because the human eye is much more sensitive to luminance changes than to chrominance changes. Next, the pixel values are grouped into 8x8 blocks which are transformed using the discrete cosine transform (DCT). The DCT yields an 8x8 frequency map which contains coefficients representing the average value in the block and successively higher-frequency changes within the block. Each block then has its values divided by a quantization coefficient and the result rounded to an integer. This quantization is where most of the loss caused by JPEG occurs. Many of the coefficients representing higher frequencies are reduced to zero. This is acceptable since the higher frequency data that is lost will produce very little visually detectable change in the image. The reduced coefficients are then encoded using Huffman coding to further reduce the size of the data. This step is lossless. The final step in JPEG applications is to add header data giving parameters to be used by the decoder.

3. Stego Encoding Experiments

As mentioned before, embedding data in the least significant bits of image pixels is a simple steganographic technique, but it cannot survive the deleterious effects of JPEG. To investigate the possibility of employing some kind of encoding to ensure survivability of

embedded data it is necessary to identify what kind of loss/corruption JPEG causes in an image and where in the image it occurs.

At first glance, the solution may seem to be to look at the compression algorithm to try to predict mathematically where changes to the original pixels will occur. This is impractical since the DCT converts the pixel values to coefficient values representing 64 basis signal amplitudes. This has the effect of spatially “smearing” the pixel bits so that the location of any particular bit is spread over all the coefficient values. Because of the complex relationship between the original pixel values and the output of the DCT, it is not feasible to trace the bits through the compression algorithm and predict their location in the compressed data.

Due to the complexity of the JPEG algorithm an empirical approach to studying its effects is called for. To study the effects of JPEG, 24 bit Windows BMP format files were compressed, decompressed, with the resulting file saved under a new filename.



Figure 2:

The BMP file format was chosen for its simplicity and widespread acceptance for image processing applications. For the experiments, two photographs, one of a seagull and one of a pair of glasses (Figure 2 and Figure 3), were chosen for their differing amount of detail and number of colors. JPEG is sensitive to these factors. Table 1 below shows the results of a byte by byte comparison of the original image files and the JPEG processed versions, normalized to 100,000 bytes for each image. Here we see that the seagull picture has fewer than half as many errors in the most significant bits (MSB) as the glasses picture. While the least significant bits (LSB) have an essentially equivalent number of errors.

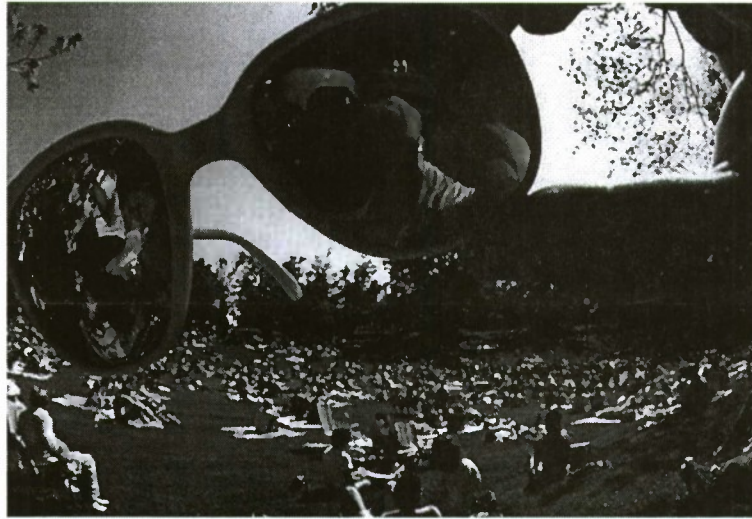


Figure 3:

	MSB 8	7	6	5	4	3	2	LSB 1
Glasses	744	4032	10247	21273	33644	42327	27196	48554
Seagull	257	991	2821	7514	15039	29941	41593	46640

Table 1:

Table 2 shows the Hamming distance (number of differing bits) between corresponding pixels in the original and JPEG processed files normalized to 100,000 pixels for each image. Again, the seagull picture has fewer errors.

	1-3	4-6	7-9	10-12	13-15	16-18	19-21	22-24
Glasses	15581	37135	30337	11976	2205	172	4	0
Seagull	24188	38710	17564	4631	409	43	1	0

Table 2:

Given the information in Table 1, it is apparent that data embedded in any or all of the lower 5 bits would be corrupted beyond recognition. Attempts to embed data in these bits and recover it after JPEG processing showed that the recovered data was completely garbled by JPEG.

Since a straightforward substitution of pixel bits with data bits proved useless, a simple coding scheme to embed one data bit per pixel byte was tried. A bit was embedded in the lower 5 bits of each byte by replacing the bits with 01000 to code a 0 and 11000 to code a 1. On decoding, any value from 00000 to 01111 would be decoded as a 0 and 10000 to 11111 as a 1. The hypothesis was that perhaps JPEG would not change a byte value by more than 7 in an upward direction and 8 in a downward direction or, if it did, it would make drastic changes only occasionally and some kind of redundancy coding could be used to correct errors. This approach failed. JPEG is indiscriminate about the amount of change it makes to byte values and produced enough errors that the hidden data was unrecognizable.

The negative results of the first few attempts to embed data indicated that a more subtle approach to encoding was necessary. It was noticed that, in a JPEG processed image, the pixels which were changed from their original appearance were similar in color to the original. This indicates that the changes made by JPEG, to some extent, maintain the general color of the pixels. To attempt to take advantage of this, a new coding scheme was devised based on viewing the pixel as a point in space (Figure 4) with the three color channel values as the coordinates.

The coding scheme begins by computing the distance from the pixel to the origin (0,0,0). Then the distance is divided by a number and the remainder ($r = \text{distance} \bmod n$) is found. The pixel value is adjusted such that its remainder is changed to a number corresponding to the bit value being encoded. Qualitatively, this means that the length of the vector representing the pixel's position in three-dimensional RGB color space is modified to encode information. Because the vector's direction is unmodified, the relative sizes of the color channel values are preserved.

Suppose we choose an arbitrary modulus of 42. When the bit is decoded, the distance to origin will be computed and any value from 21 to 41 will be decoded as a 1 and any value from 0 to 20 will be decoded as a 0. So we want to move the pixel to a middle value in one of these ranges to allow for error introduced by JPEG. In this case, the vector representing the pixel would have its length modified so that the remainder is 10 to code a 0 or a 31 to code a 1. It was hoped that JPEG would not change the pixel's distance from the origin by more than 10 in either direction thus allowing the hidden information to be correctly decoded.

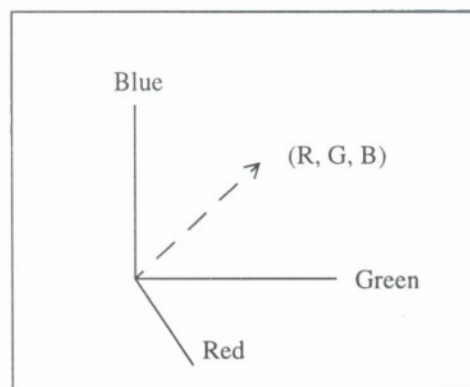


Figure 4:

For example, given a pixel (128, 65, 210) the distance to the origin would be computed: $d = \sqrt{128^2 + 65^2 + 210^2} = 254.38$. The value of d is rounded to the nearest integer. Next we find $r = d \bmod 42$, which is 2. If we are coding a 0 in this pixel, the amplitude of the color vector will be increased by 8 units to an ideal remainder of 10 ($d = 262$) and moved down 13 ($d = 241$) units to code a 1. Note that the maximum displacement any pixel would suffer would be 21. Simple vector arithmetic permits the modified values of the red, green, and blue components to be computed. The results of using this encoding are described in the next section.

Another similar technique is to apply coding to the luminance value of each pixel in the same way as was done to the distance from origin. The luminance, y , of a pixel is computed as $y = 0.3R + 0.6G + 0.1B$ [6]. Where R , G , and B are the red, green, and blue color values respectively. This technique appears to hold some promise since the number of large changes in the luminance values caused by JPEG is not as high as with the distance from origin. One drawback of this technique is that the range of luminance value is from 0 to 255 whereas the range of the distance from origin is 0 to 441.67.

4. Discussion of Experiments

With ordinary embedding techniques which simply replace image bits with data bits one is forced to use an enormous amount of redundancy to achieve a suitably low error rate after JPEG compression. Also, since the lowest few bits are so badly scrambled by JPEG, higher order bits must be used which increases the visible distortion of the image. This is contrary to steganography's goal of being a covert communication channel.

The distance from origin technique results in a lower error rate, but requires the addition of repetitive redundancy to even attempt to achieve a reasonable error rate. Specifically, the average displacement by JPEG of pixels in the seagull picture with respect to the origin is 2.36. But, if a modulus of 62 is assumed, the number of pixels displaced by 30 (enough to cause an error) is 3100 or 0.8678%. Given this figure and applying triple redundancy in embedding data (i.e. embed each bit three times in a row) yields a theoretical error rate of $(.008678)^3 + 3(0.008678)^2(0.991322) = 0.000225$ per bit or $1 - (1 - 0.000225)^8 = 0.001799$ per byte.

Unfortunately, these calculations do not take into account the peculiar and obstreperous nature of JPEG processing. JPEG produces the most errors in areas of an image where the pixel values vary the most, i.e. color transients produce local errors the number of which is proportional to the amount of the transients. The process of embedding data invariably causes a different pattern and amount of color transients. So embedding data actually *causes* more errors in the area where data is embedded.

Despite the active way in which the JPEG algorithm garbles embedded data we were able to demonstrate a recovery rate of approximately 30% of embedded data using RGB coding coupled with multiple redundancy coding.

5. Summary

As an anti-steganography technique, JPEG is very effective. In order to achieve anything approaching an acceptable error rate, a great deal of redundancy must be applied. With the distance-to-origin and luminance modulo coding techniques the error rate can be brought down. But these techniques must be coupled with multiple redundancy to lower the error rate. Although the amount of data which can be hidden may be relatively small compared to the size of the image, the security threat that steganography poses cannot be completely eliminated by application of a transform-based lossy compression algorithm.

This paper describes preliminary research. Future work will examine the internal details of the JPEG algorithm to determine the effects of the discrete cosine transform and the quantization factors on the information content of images with and without steganographic embedding. We are also investigating potential techniques to detect steganography in images.

6. Acknowledgements

The authors would like to thank Hannelore Campbell, Hal Fredericksen, and David Wootten for many helpful ideas, criticisms, and discussions.

References

1. Bradley, J., XV - Version 3.00, Rev. 3/30/93.
2. Cha, S.D., Park, G.H., and Lee, H.K., "A Solution to the On-Line Image Downgrading Problem," in *Proceedings of the Eleventh Annual Computer Security Applications Conference*, New Orleans, LA, pp. 108-112, December 1995.
3. Kahn, D., *The Codebreakers*, MacMillan Company, 1967.
4. Kurak, C., McHugh, J., "A Cautionary Note on Image Downgrading," in *Proceedings of the 8th Annual Computer Security Applications Conference*, pp 153-59.
5. Wallace, Gregory K., "The JPEG Still Picture Compression Standard", *Communications of the ACM*, Vol. 34, No. 4, April 1991.
6. Pennebaker, William B., Mitchell, Joan L., *JPEG Still Image Compression Standard*, Van Nostrand Reinhold, New York, 1993.
7. *Joint Photographic Experts Group (JPEG) Compression for the National Imagery Transmission Format Standard*, MIL-STD-188-198A, December 1993.

KEY ESCROWING SYSTEMS AND LIMITED ONE WAY FUNCTIONS

William T. Jennings
Southern Methodist University
& Raytheon E-Systems,
Greenville Div.

Phone : (903) 457-6756
FAX : (903) 457-7640
E-Mail : wtj1@esygv1.com

James G. Dunham
Southern Methodist University
Dallas, Tx. 75275-0335

(214) 768-3484
(214) 768-3883
jgd@seas.smu.edu

ABSTRACT

The topic of key escrow has received considerable attention in the literature as of recent. One problem with existing public proposals for multi-agency secret splitting is that they do not address concerns that individuals within those agencies might work in collusion to gain access to large amounts of valuable keying information. This work suggests a solution to prevent large scale, random abuse of privilege. The basis of this proposal is to add a limited one way work function to make the withdrawal process much more difficult than the deposit process, thus limiting the ability to make excessive numbers of withdrawals.

KEYWORDS: Key Escrow, Algorithm, Clipper, Limited One Way Functions

KEY ESCROWING

The term *Key Escrowing* has recently emerged in the literature in reference to systems which are intended to provide the capability for cryptographic key storage and retrieval. The consideration of such systems was largely ignored in the literature until controversy arose over recent government proposals concerning public cryptographic standards.

In April, 1993, the Clinton Administration made a public proposal suggesting a NSA designed encryption/decryption cryptographic device, designated the Clipper Chip, to be made available for private sector use [1]. Attacks on networked computer systems reported are growing exponentially [2]. Additionally, the introduction of digital technologies coupled with the availability of private data encryption has reportedly made the task of legal wiretapping by law enforcement agencies very

difficult to perform. The introduction of the Clipper hardware is an attempt to solve both problems. The proposal, however, has met with a great deal of criticism [3, 4]. Included in the proposal was a requirement that the master keys for these devices be registered with the government. The registration has been referred to as Mandatory Key Escrowing (MKE).

As a consequence of the legal implications of MKE, it has been suggested that separate agencies or agents would be given separate components of the key. This would be accomplished using secret splitting techniques [5]. Only when the components are put back together may the key be successfully recovered and used. Each of these agencies would serve in the role of a trusted authority or "escrow agent" [1].

A *Key Escrowing* system is fundamentally different from cryptographic systems based on zero-knowledge techniques such as some password or authentication systems. A Key Escrowing system must provide a withdrawal capability rather than simply verification. The value of the keys stored is equal to the sum of all the contained keys. Thus for a national key escrow system, the economic value of stored keys would be immense. There are interesting technical questions that are raised. Barlow [1] has raised the issue of whether or not key depositories would themselves become the target of criminal or terrorist organizations, raising the uneasy question of what happens if the key depositories themselves are compromised. The key depositories, containing an enormous wealth of information, would serve as a high priority target for terrorists or hackers. Someone with access to keying information stored in large key escrow databases would be confronted with an tremendous temptation to browse through all therein. Current proposals would require collusion between escrow agencies or individuals within those agencies in order for abuse to occur. It is possible to envision circumstances whereby this might happen.

One response is to simply argue that key escrowing should not occur. It is however entirely likely that it will occur anyway. There are of course secret sharing schemes that are more elaborate and might require larger groups or more agencies to act in collusion to reveal the secrets. Creating more agencies and splitting the information up into more pieces does seem to provide a higher level of security. While it requires more individuals to act in a collusive manner, it does not change the basic nature of the problem's weakness.

It is apparent that, for such a system, what may be desired is a self-limiting or self-regulating algorithm to preclude the potential for wholesale abuse. Indeed this same argument has been offered by Bellare and Goldwasser [6]. Since keys are very frequently created (deposited), but rarely withdrawn, there is an inherent asymmetry to the problem that can be used to advantage. If it were as easy to withdraw keys as it is to deposit them,

then there is every possibility that keys may be withdrawn at an excessive rate. Therefore it would be preferable to design a system that inherently limits the rate of withdrawals to a pre-defined maximum rate. This provides a deterrent against a specific threat profile, that of the "casual key browser," or systematic abuser.

This application thus suggests that there should be a specific cost associated with each key withdrawal from a Key Escrowing depository. This then means that one may not simply randomly browse among what is in principle a very large data base of keys but must in general request specific keys to be withdrawn. What is proposed is a function designed in such a manner that the computational cost of key withdrawal greatly exceeds the cost of deposit in a controllable manner. Such a proposal may not in itself prohibit an authorized individual from asking for additional keys as well, but the numbers would be inherently self limiting. Additionally any continuing pattern of such behavior would be statistically detectable, since the cost (on average) for systematically requesting additional keying information would be detectable.

This subject is of wider interest than that of simply addressing issues concerning the government sponsored Clipper chip. There are indeed important commercial applications for Key Escrowing systems as well. Valuable corporate information that should be protected by strong cryptographic methods often is not. Recall that data thus encrypted is totally unrecoverable without the keys. Therefore it is highly desirable that commercial systems provide key escrowing capabilities to facilitate data recovery in the event that keys are lost.

KEY ESCROWING SYSTEM IMPLEMENTATION

Traditional electronic approaches to the storage of vital key information normally involve keeping copies of keys in a trusted database, protected by one or more master keys. This master key is therefore more valuable than the other keys and is at least as important as the sum of all of the keys that it protects. Consequently, anyone in possession of a master key would be afforded complete and unlimited access to all of the information protected by all of the sibling keys stored using the master key. The advantage of having a master key is that there is only one key of which to keep track. The chief disadvantage is that a master key constitutes a single point defense. Compromise of a such a master key is therefore very critical. A single key database can be compromised by anyone in possession of the master key material.

It therefore would seem to be desirable to segregate data into multiple master key domains. This of course has the undesirable property of multiplying the number of master keys which must be safeguarded or protected in of themselves. Ultimately these keys

are protected in much the same manner as a single master. It also perhaps suggests that hierarchical approaches suffer similar maladies and do not really address the underlying problem.

There have been suggestions that solve this problem by using secret splitting techniques to provide complementary components for each key stored. These components would have to be put together to recover the original key. Components would be separated at time of creation and stored with alternate "trusted" agencies. These techniques not only offer protection from external attack on the database but also some protection from abuse from within a particular trusted agency. These ideas are inadequate in that they do not address concerns over the possibility of collusion between individuals within the agencies with access to the databases. It is recommended that additional measures are necessary to discourage abuse of the system and to provide additional opportunities for oversight.

KEY ESCROW AND LIMITED ONE-WAY FUNCTIONS

Key Escrowing systems can be characterized as being strongly one way in their basic input/output bandwidth requirements. Many keys are created, but few are ever retrieved. Typically the input bandwidth far exceeds the aggregate output bandwidth, perhaps by many orders of magnitude. A balanced design for such a system might suggest that the algorithm for storage and retrieval match the actual bandwidth requirements. It would be advantageous to implement an algorithm that requires far less work to make a deposit than it does to make a withdrawal. It is proposed herein to refer to applicable functions that display asymmetric work requirements as being *limited one-way functions*. This proposed methodology is to use limited one-way functions to effectively limit the rate of withdrawals. We draw the distinction from normal one-way functions and hence use the term *limited* because we want to look at candidate functions which are not necessarily strongly one-way.

Candidates for useful functions should be provably asymmetric. Ideally there should be provable bounds on the ratio of the amount of work required to go forward versus the work required to go back. This is important because the effectiveness of the ability of the function to impose cost on the user is characterized by the upper and lower bounds on this ratio. Another aspect of this methodology deviates significantly from a classical cryptographic application. Since the key escrow database server has access to the plaintext key information by possessing the master key, but is simply being penalized by a work function for key withdrawal, the algorithm may legitimately only require that each transaction be accomplished taking a prescribed length of time, on average. This constitutes a significant shift of paradigm. The concept of the penalty is to limit or regulate the general flow of data out of the key escrow database. Hence, to satisfy the demands of this requirement, it

may only be necessary to determine the average or statistical complexities of the limited one-way function and it's inverse.

One possible candidate would simply be to use a suitable cryptographic technique with a limited key size. This is the most straight-forward approach. Conceptually it is very similar to partial key escrow techniques such as proposed by Shamir [7]. The difference in this case being that the entire key may be escrowed but the work may be imposed prior to accomplishing key withdrawal rather than after withdrawal from the escrow. The decryption (withdrawal) is accomplished either by brute force techniques or by directly breaking the key. Since suitable cryptographic techniques to accomplish this are based on solving NP-complete problems, there are not provable tight lower bounds on the work required to accomplish this. Additionally there may be a large differential in work required between the normal withdrawal technique (if implemented by brute force) and the backdoor path (breaking the key). Therefore there are not necessarily very tight controls on the work required to accomplish this.

We propose another example of how a suitable Limited One Way function might be implemented. The following is an extension to an algorithm originally proposed by Merkle [8]. Let us consider a case where we shall define (following Merkle's original terminology) the puzzle transmitted by Alice to Bob to be as follows: Alice generates, using the encryption keys, matched cryptogram/decryption key pairs (C_i, Kp_i) , $i = 1, 2, \dots, N$ corresponding to a set of messages $\{M_i\}$ $i = 1, 2, \dots, N$ where i is simply an index used to identify which member of the set of pairs is referenced. The message M_i contains a corresponding token T_i . In this example the familiar RSA system is used to illustrate the concept. This is not however a general requirement. Thus Alice generates the puzzle set:

$$P = \{(C_0, Kp_0), (C_1, Kp_1), \dots, (C_i, Kp_i), \dots, (C_N, Kp_N)\} \quad (1)$$

where C_i is the i th cryptogram corresponding to the i th message M_i , and where Kp_i is the i th public key generated by Alice. Kp_i is used to encrypt M_i and corresponds to Ks_i which is the secret key retained by Alice. It is assumed that they share the commonly agreed upon encryption function. Alice communicates the set P to Bob.

Bob selects j at random, where $j \in 1, \dots, N$, then chooses the j th ordered pair, (C_j, Kp_j) , from the set P . Bob derives T_j from C_j by performing the decryption:

$$D(Kp_j, C_j) = M_j; T_j \subset M_j, \quad (2)$$

where D is the decryption function.

Bob then forms the message $\mu_j = (T_j \ \&\& \ R)$, the concatenation of the selected token and a randomly chosen vector R . Bob proceeds to form the response message S , such that:

$$S = E(Kp_j, \mu_j), \quad (3)$$

and sends S to Alice. Alice may then recover T_j by application of the secret key Ks_j . Alice does not know which of the N keys to use and thus must try keys randomly from the set of N until a match is made.

It is assumed that the channel between Alice and Bob cannot be tampered with but is not secure. An observer, Carol, may see both the initial message P and the response S from Bob. Carol therefore has all of the N public keys but does not have the corresponding secret keys. To "discover" the message Carol is faced with the problem of first deriving the N tokens, then forming $N \cdot R$ messages of the form $(T_j \ \&\& \ R)$. Finally Carol must then encrypt these and compare the result to S in order to discover Bob's choice for j .

We should consider the amount of work imposed by this algorithm upon the various parties involved. The work that Carol is forced to perform is now greater however than that performed by either Bob or Alice. Carol does not have the benefit of having the decryption keys that are available only to Alice. Carol must try all $\text{Avg}(N \cdot R)$ possibilities to discover the decision that was derived where we use the notation $\text{Avg}(\)$ to refer to the average complexity. Refer to this approach for obtaining the solution as the "front-door" approach.

Carol is at a disadvantage to Alice by a factor of $\text{Avg}(R)$, the amount of randomization information embedded in the problem. This is because Carol does not possess the decryption keys which are the sole property of Alice and are not revealed in the process. Carol is forced to try all $\text{Avg}(N \cdot R)$ combinations until a match is found. Carol does, however, have an alternative possible attack. Carol may attempt to break $\text{Avg}(N)$ decryption/encryption key problems, directly attempting to discover the secret keys. This approach to solving the problem is referred to as being the traditional "back-door" approach to solving the problem. The work associated with this approach thus represents an upper limit on the amount of work that Carol must perform. System parameters thus can be chosen such that Carol is forced to go in through the built in front door, because that is the only computationally viable path. Let the amount of work performed to directly break the key problem by brute force methods (the back-door approach) be represented by $\text{Avg}(W_B)$. Let the amount of work that Alice performs using trapdoor information to accomplish a decryption be represented by $\text{Avg}(W_T)$. We shall presume that for reasonable choices of system parameters that $\text{Avg}(W_T) \ll \text{Avg}(W_B)$. We can also reasonably presume that $\text{Avg}(W_T) \approx W_E$ if the encryption and decryption processes are

symmetric. This assumption is true for instance of some public key cryptosystems such as RSA. The work that is now required by each party involved is given by:

$$W_{\text{bob}} = W_D = W_E, \quad (4)$$

$$W_{\text{alice}} = N * W_E = \text{Avg}(N * W_T), \quad (5)$$

$$W_{\text{carol}} = \text{Avg}(N) * W_D = \text{MIN}(\text{Avg}(N * R * E), \text{Avg}(N * W_B)). \quad (6)$$

The work required by Bob to efficiently perform this calculation (assuming RSA) can be estimated to be $Kn^2 \log n \log \log n$, where K is a system dependent constant [9]. It was recently reported that the fastest single chip implementation for performing modular exponentiation is capable of evaluating 560 bit operations per 5.5 msec [10]. Consider an example system using this chip, using 560 bit numbers and taking N to be 10^3 and taking R to be 10^5 .

$$W_{\text{bob}} = W_D = 5.5 \text{ msec} \quad (8)$$

$$W_{\text{alice}} (\text{avg}) = N * W_E / 2 = 2.75 \text{ sec} \quad (9)$$

$$W_{\text{carol}} = N * R * W_E / 2 = 2.75 \times 10^5 \text{ sec} \approx 3.2 \text{ days}. \quad (10)$$

By using this method it is possible to control the amount of work that Carol must perform to solve the puzzle. In the example above, withdrawals could only occur in this system at the maximum rate of about 114 per year. This is reasonable assuming about 10 regional depositories. The number of court-ordered wiretaps for all federal, state, and local law enforcement purposes is approximately 1000 per year. Specifically there were 919 wiretaps authorized in 1992 and 976 in 1993 [5]. Carol is forced to perform a very large number of simple operations (on average) to resolve the answer. Because of this it is possible, by adding enough randomization, to take advantage of average computational complexity in determining the required work. This has a distinct advantage over implementing a single weak cryptofunction such as with a limited key size. The desired performance of the proposed algorithm can be controlled by adjusting the statistical parameters. This offers a greater degree of control over the results than that offered by the simpler approach.

To apply this algorithm to the problem of Key Escrow, we can consider a record made of the exchange between Alice and Bob (such as would be seen by Carol) as the material to be deposited in the escrow. In this scheme, Bob and Alice negotiate for a key exchange with Alice as the key requester and Bob as the key generator. Carol represents the recording/withdrawal mechanism. Prior to storage, the transaction is encrypted using a strong cryptographic technique and master keys used to protect the overall database. It is also practical to incorporate secret splitting mechanisms as well. Depending on the application Alice may either keep her secret puzzling keys or they may simply be discarded as part of the process. This escrowing process is illustrated in Figure 1.

Withdrawal of the keying material would involve retrieval of the transactions that had occurred between Bob and Alice first using the database master key for decryption to recreate the transaction. This transaction would then have to be "broken" in the manner that Carol would need to accomplish in order to discover Bob and Alice's agreement. Thus this second stage of decryption represents the controllable work function used to limit the rate of key withdrawal. This withdrawal process is illustrated in Figure 2.

SUMMARY

Key Escrowing Systems have unique characteristics which distinguish them from other cryptographic systems. To address some of the unique requirements of these systems, the concept of limited one-way functions was introduced. This proposed technique is not intended to replace master keys or secret splitting techniques intended to preserve integrity of the data from external attack. Instead this additional layer of protection is intended to limit the ability of otherwise properly authorized individuals to withdraw keys at an excessive rate. An example algorithm was also introduced. It was suggested that this concept is a new tool available to deal with situations where the rate information retrieval is desired to be controlled or where some minimum time limit is to be asserted within a defined probability. This technique may be applicable to problems other than Key Escrow as well.

An additional layer of functionality is provided by the proposed algorithm in the form of a specific work function and hence economic costs associated with the function of key recovery from a key database. Master keys can still be used to provide the front door into the main database. Additionally this technique does not preclude the possibility of also using secret spitting techniques as well. The added value is that once inside the door, there is no free access to any and all information contained therein. The data contained can only be obtained by an authorized individual who in addition can afford to pay the price of retrieval. The specific benefit of this approach is that an inherently limiting or regulatory process is imposed. This means that general abuses can be limited, thus solving a fundamental problem that is not addressed by conventional cryptographic methodologies.

In a Key Escrowing system, exemplified by that proposed for a national communications system, inclusion of a methodology for limiting withdrawal bandwidth would provide the ability to prevent collusive parties from freely shopping among the keying information without imposed constraints. The system could be designed so that the withdrawal rate was adjusted such that only a reasonable number of withdrawals over a period of time would be possible. What is provided however is a greater level of privacy protection to the general populace from the possibility of widespread random monitoring of otherwise private transactions.

REFERENCES

- [1] J. P. Barlow, "A Plain Text on Crypto Policy," *Communications of the ACM*, Vol. 36, No. 11, Nov. 1993, pp. 21-26.
- [2] S. Staniford-Chen, L. T. Heberlein, "Holding Intruders Accountable on the Internet," *Proceedings of the 1995 IEEE Symposium on Security and Privacy*. May, 1995, pg. 39-49.
- [3] R. W. Holleyman, "On the Export of Software with Encryption Capabilities", testimony presented at Key Escrow Meeting, NIST, Gaithersburg, Maryland, Sept. 6 1995.
- [4] U. S. Senator Patrick Leahy, "Statement of Patrick Leahy on Vice President Gore's Clipper Chip Letter," public letter, 21 July 1994.
- [5] Froomkin, Michael A., "The Metaphor is the Key: Cryptography, the Clipper Chip, and the Constitution," *U. Penn. Law Rev.* 709, 1995.
- [6] Denning, D. E., "Descriptions of Key Escrow Systems" unpublished, <http://www.cosc.georgetown.edu/~denning/crypto/Appendix.html> reference to technique attributed to Bellare and Goldwasser.
- [7] Denning, D. E., "Descriptions of Key Escrow Systems" unpublished, <http://www.cosc.georgetown.edu/~denning/crypto/Appendix.html> reference to technique attributed to A. Shamir.
- [8] Merkle, R. C., "Secure Communications Over an Insecure Channel," *IEEE Trans. on Information Theory*, 1976, IT-22, pp. 644-654.
- [9] Borodin, A., "Computational Complexity: Theory and Practice" from *Currents in the Theory of Computing*, Aho, A.V. editor, Prentice Hall, 1973.
- [10] Orup, H., "Simplifying Quotient Determination in High Radix Modular Multiplication," *Proceedings of the 1995 IEEE 12th Symposium on Computer Arithmetic*, July 1995, pp. 193-199.

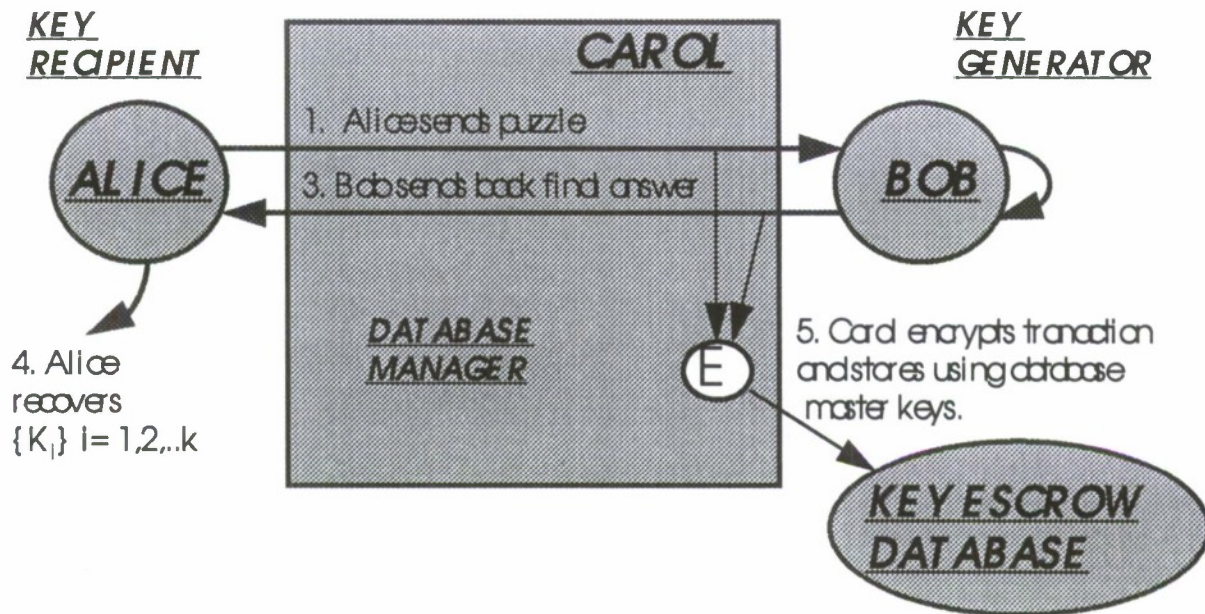


Figure 1 - Key Escrow Process

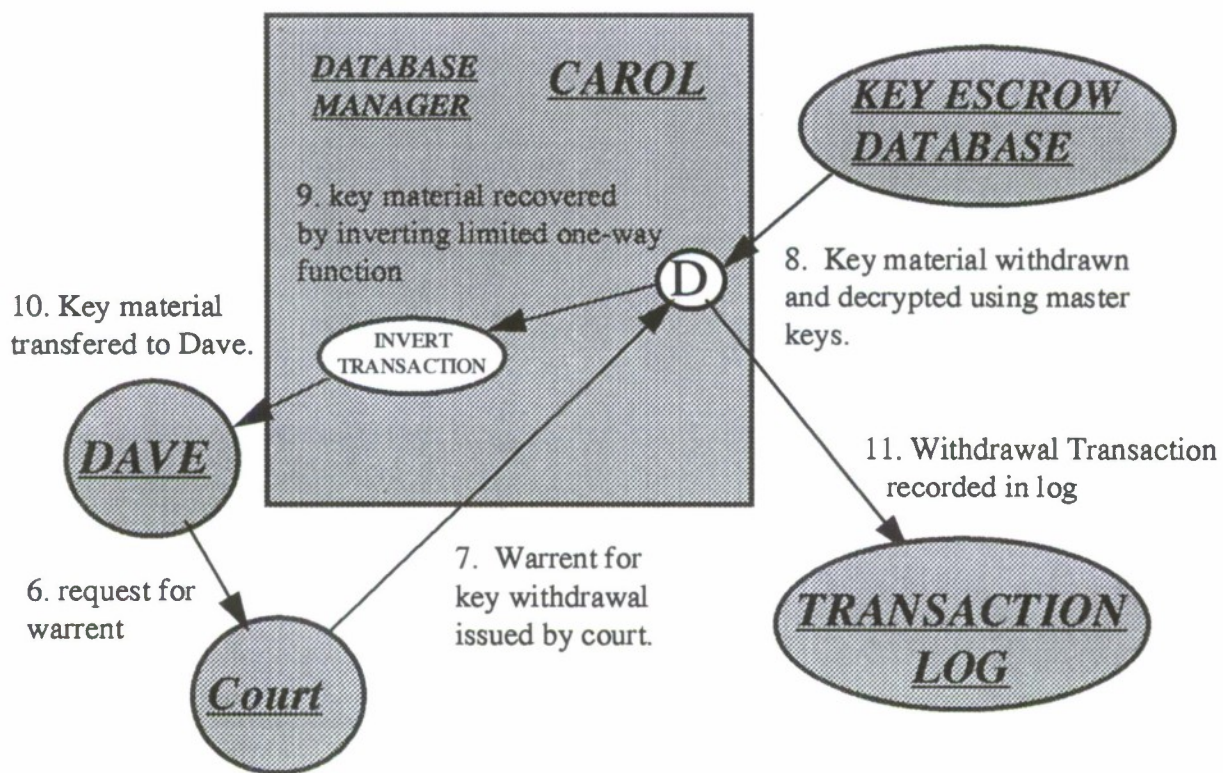


Figure 2 - Key Withdrawal Process

Key Escrowing Systems and Limited One Way Functions

William T. Jennings
Southern Methodist University
& Raytheon E-Systems, Greenville Div.
(903) 457-6756
wtj1@esygv1.com

KEY ESCROW SYSTEMS

- Only legitimate to make withdrawals one at a time.
- Should never need to read entire database.

KEY ESCROW SYSTEMS

- Requires strong encryption protection
- May require secret sharing/splitting techniques

KEY ESCROW SYSTEMS

Characterized by
Many Deposits, Few Withdrawals

Write BW >> Read BW

KEY ESCROW SYSTEMS

- Requires Central, Trusted Authority
- Requires strong encryption protection
- May require secret sharing/splitting techniques

PROPOSED ALGORITHM

- Example of Limited One-Way Function.
- Alice generates pairs of cryptograms and decryption keys.
- Each cryptogram conveys a token.

LIMITED ONE WAY FUNCTIONS

- Feasible but costly to invert.
Requires both time and computational resources.

PROPOSED ALGORITHM

- Bob selects one at random and performs the decryption.
- Bob takes the token, adds randomization and signature information, then re-encrypts.
- Alice uses retained secret keys to recover token.

PROPOSED ALGORITHM

- Carol represents a passive observer of the channel recording the transaction.
- Withdrawals occur by either breaking the underlying cryptosystems (intractable), or solving large numbers of simple puzzles (built-in front door).

Why Apply Limited One-Way Functions?

- Addresses class of problems not adequately resolved by conventional cryptographic means.
- Withdrawal bandwidth can be adjusted to fit available bandwidth.
- Built In Work Function discourages excessive withdrawals. Makes abuse infeasible provided resources physically limited.
- Patterns of abuse may be detectable due to systemic (measurable) use of resources.

The Keys to a Reliable Escrow Agreement

Richard Sheffield
3539 A Church St.
Clarkston, GA 30021-1717
1-800-875-5669
escrow@mindspring.com

*"Forethought in any part of life is seldom regretted
and always rewarded."*

Perhaps one of the most admired industries in the world is that of computer software. Most companies who comprise this industry are looked upon as rapidly growing, innovative enterprises. These companies are often admired as smaller operations that must fight to stay on top in a highly competitive industry. Yet these are the very characteristics that scare so many of the clients of these developers. An end-user of the software often wonders, and rightly so, that if the software developer fails to prevail in this competitive industry, if bankruptcy is only a few months away. *The Economist*¹ has reported that as many as 60% of all newly created high-technology companies will disappear within five years. The recognition of the corporate mortality of technology developers extends beyond this industry. This concern has also alarmed business partners of technology companies who might be involved in co-development projects. Lastly, investors, such as venture capitalist, are looking for ways to secure their interest in proprietary information.

Background

These are the concerns that gave birth in the late 1970s to *software escrow services*. These arrangements require the developer of a software product to place proprietary materials necessary to maintain the product in escrow with a neutral party known as the *escrow agent*. Should the software

vendor, or *licensor*, fail to support the product, the escrow agent agrees to release the proprietary materials (such as source code) to the end-user. The end-user, or *licensee* is then allowed to employ the deposit materials to support the licensed product. In theory, the service seems reliable and should allow the licensor and its client to proceed with their business relationship. However, does the software escrow service really protect the licensee or does it only provide an erroneous sense of security?

The opponents of these arrangements turn to the examples of failed escrow agreements as evidence that the concept is a flawed one. All too often, the parties involved in licensing negotiations rush through the establishment of the escrow and neglect the issues that distinguish an effective escrow from a valueless service. Todd Volyn, an attorney with Shell Oil Company notes, "Raising the point [of escrow] threatens to considerably lengthen the negotiation process - an outcome that nobody desires. Thus, the parties come to a raw compromise without considering the cause of the concerns, whether an escrow is even capable of meeting those concerns, and alternative solutions."² In particular, the conditions upon which the escrowed materials may be released are often so restrictive, that it is not possible for the licensee to retrieve the escrow documentation, even if the developer has clearly failed in its responsibilities. Moreover, if the licensee is successful in acquiring the source code and other materials, it may have been so poorly prepared that it is useless in the efforts of licensee to assume the responsibility for the maintenance of the licensed program.

While these criticisms are fair ones, they do not indict the entire escrow process but rather point to the lack of foresight in the creation and execution of many escrow agreements and services. When properly prepared and executed, an escrow arrangement will provide a safety net under the relationship between the parties involved in technology transfers.

A Prepared Escrow Policy

Most companies have no established escrow policy, leading to uncertainties within an organization of when to request an escrow from software vendors and what type of service should be required. Many licensees go to the trouble to demand an escrow from their vendor, but the decision of who to escrow the materials with is left up to the vendor. Not surprisingly, the vendor usually opts for the least expensive service, such as the vendor's own attorney. While an arrangement of this nature may provide a quick solution, it often sacrifices the security of the escrow. Many contracts of this kind include insurmountable obstacles to the licensee's acquisition of the deposit materials. For example, the only condition that the material may be released will be "complete corporate dissolution" of the software vendor, a condition that omits service faults, reorganization or the transfer of ownership of the vendor.

For the company that licenses software and is unable to acquire the proprietary information central to the product, a standardized policy of when to ask for an escrow and what type of service is required will often ensure a successful escrow arrangement. The licensee's counsel or legal department should coordinate such a policy with the purchasing and Information Technology (IT) group within the company. This strategy must offer the purchasing agent criteria for evaluating if the developer is likely to fail in support of the product. The escrow requirement should be submitted before the license agreement is executed. This provides the licensee with significantly more leverage in its request and allows portions of the escrow agreement to be referenced in the license agreement.

The Escrow Agreement

The foundation of any escrow service is the agreement. Although many escrow agents will offer their own form contract, the vendor and licensee may

both profit by having their own terms to present to one another when beginning the escrow discussions. Counsel for either side should be all too familiar with what they and the opposing side require of the escrow, particularly the release conditions which are at the heart of the contract. An attorney representing the licensee is likely to offer liberal release conditions that allow the licensee to receive the deposit materials immediately upon any request. Conversely, as mentioned above, the developer's counsel will insist upon a release only in the event of the vendor's dissolution as a corporate entity and require the licensee to pay all fees associated with the service. While a truly neutral escrow agent is unlikely to take a position on such issues, one may expect the agent to offer suggestions based on their experience and present their own requirements of the contract. The agent is likely to insist on the clarity of such instructions that dictate the terms of release. Vague release requirements benefit no one and often invite extended court battles.

The effectiveness of the service is determined by a number of other sections of the agreement. First, the contract should establish a date by which the vendor is required to deliver the deposit materials to the agent. Often an agreement is reached to escrow the materials with an agent, but none of the parties are responsible for ensuring the materials are actually delivered. Second, the developer may want to consider placing restrictions on the licensee's use of the deposit materials should they ever be released. Such conditions usually correspond to the constraints found in the license agreement. Finally, there is the question of the indemnification and liability of the agent holding the materials. It is in these sections that the escrow agent is likely to offer its strongest opinion, insisting on reasonable limits to its liability.

The Deposit Materials

The escrow agent is likely to offer great insight in how to prepare an effective escrow deposit. The deposit should be made within a reasonable period following the execution of the escrow agreement. A

period of 15 to 30 business days is typically agreeable to both sides. Should there be a delay in this deposit, the escrow agent should notify the licensee of the developer's failure to comply with the agreement. This delay may be due to a number of factors, including the developer's intentional noncompliance with the agreement or something as innocent as a delay in the scheduled development of the product. However, even if the product is not fully developed at the time the deposit is required, a partial deposit likely benefits the licensee more than a hollow arrangement with no deposit. If the delay is due to the developer's oversight or disregard for the arrangement, the licensee is usually the only party with enough leverage to influence the developer to make the deposit. Ultimately, the developer should work to make both the initial and update deposits in a timely manner. All parties might find it helpful to agree upon an established schedule, such as quarterly updates, to ensure the escrow deposit remains up to date.

While some critics of an escrow cite incomplete deposits as evidence of the ineffectiveness of an escrow, it is not an inherent problem of the service. In fact, deposits that are of no ultimate use to the licensee are often the result of failed communications between the parties as to what materials would constitute a usable escrow deposit. It is common for the parties to hurry through this discussion and to disregard the advice of the agent who likely has some valuable background as to what materials have historically made up a reliable deposit. In any event, communication on this matter between the agent, a programmer for the developer, and the ultimate end-user for the licensee will go a long way towards the creation of a reliable escrow deposit. Many larger corporations have found it useful to create a standardized list of escrow materials that are required from all developers. This policy insures that all escrows created will have a comprehensive deposit. It also prevents key materials from being omitted on account of an accidental oversight by a hurried developer, or the negligent action of a careless employee on either side. A partial list of recommended deposit materials includes:

- Two copies of the Source Code for each version of the licensed software on magnetic media
- All manuals not provided to the licensee (technical, operator/user, installation)
- Maintenance tools and necessary third party system utilities
- Detailed descriptions of necessary non-liscenser proprietary software, descriptions of the programs required for use and/or support that the developer does not have the right to offer to the licensee
- Names and addresses of key technical employees that a licensee may hire as a subcontractor in the event the developer ceases to exist
- File listings generated from any magnetic media
- Compilation instructions in written format or recorded on video format.

When the materials have been prepared by the developer, they should be shipped to the escrow agent through a traceable courier. Included in this shipment should be a complete set of the materials, a letter from the developer certifying the accuracy of the materials, and an inventory list of each item of the escrow deposit. Upon receipt its receipt of the materials, the agent should contact both the developer and the licensee to verify the material's arrival. This notification should be sent to an individual determined to be the primary contact for all activity relating to the escrow account.

Storage of the Deposit Materials

Unlike most paper documents, magnetic media requires unique storage conditions. It is critical that

both the developer and the licensee be familiar with the escrow agent's storage facilities before contracting for this service. It is common for the company who agrees to store magnetic media to also store other valuables, such as jewelry or personal collectibles. This practice is problematic for the developer and licensee, because these types of facilities are more susceptible to burglaries and vandalism. If possible, an agent of the licensee and developer should confirm that only magnetic media or technology documentation is being kept at the storage location.

An environment designed to store paper documentation, such as a safety deposit box or safe, differs from the environment required to store magnetic media. Because fluctuating temperatures and humidity can damage the media, the escrow agent should have a *media vault* in which to secure the deposit materials. These vaults avoid dangers to these types of materials. For example, a media vault will not have standard water fire extinguishing systems which, if activated, will destroy magnetic media. An inspection of such facilities should confirm these qualities in an escrow agent's facilities:

- fire retention walls with a minimum four hour fire rating
- Halon or some alternative gas fire extinguishing system
- storage environment that maintains constant temperature and humidity
- extensive security systems shielding the facilities

The escrow agent should be required by the developer to secure the confidentiality of the media. The developer often receives commitments from the agent that the material will be not be available to any party once it is delivered to the agent, except as required by the agreement. To ensure this condition is met, the developer may wish to seal the materials in a package that will remain unopened when the material is

received by the agent. Periodic audits of the facility and condition of the materials should help assure the safe keeping of the escrow deposit. During these audits, one should examine the insurance held by the agent and look for coverage designed for escrow services, such as liability and errors & omissions coverage.

From time to time the product being licensed by the end-user is upgraded. It is critical that major updates to the source code be sent to the escrow agent so that the escrow deposit correctly corresponds to the version being used by the licensee. Otherwise, if the escrow is ever released, the licensee may find the escrow deposit to be antiquated and therefore useless for its purposes. While the primary responsibility for shipping upgrades to the escrow agent lies with the developer, both the agent and licensee may play an active role in updating the escrow deposit. For example, quarterly phone calls may be made by the agent to each party asking if an update has recently been shipped to the licensee and if such an update should be made to the escrow. The involved parties should decide whether the older materials should be returned to the developer or continued to be stored when an update is made. Most escrow agents recommend the continued storage of at least one past version. This practice protects the licensee in the event that there is ever a problem reading or using the newer version and serves the developer by documenting the development date of each update. Finally, because magnetic media degrades over time, the escrow agent should require the developer to ship a new copy of the materials every three years if the materials are never updated.

Verification of the Deposit

Many licensees are understandably concerned about the accuracy and reliability of the escrow deposit. Frequently a licensee will find that when the deposit is released, it was prepared so poorly by the developer that it is useless for supporting the licensee's system. Critics of software escrow point to these cases as the preeminent obstacle to a successful escrow

agreement. However, there are several solutions to this potential shortcoming.

The first step towards securing confidence in the deposit is to be attentive during the period the initial deposit is being prepared. All too often, the escrow deposit is hurriedly compiled and critical components are omitted. By simply showing interest in what materials are shipped by the developer to the escrow agent, the licensee will increase the likelihood that the deposit will be functional. Ideally, the escrow agent should coordinate discussions between each party involved in the matter. Together, each side can contribute their ideas as to what materials will make the escrow effective. A list of materials is then compiled and used by the developer to gather the materials for shipment to the agent. At this point, the escrow agent must visually verify the materials that it receives. This procedure will act as a second line of defense against an incomplete deposit. Lastly, the licensee should reserve the right in the escrow agreement to test any updates.

The option of validating the escrow deposit is the least used but most effective way of ensuring effective escrow protection to the licensee. The more capable escrow agents provide a testing service to their clients, usually through an in-house agent or a professional software testing agency. One such agency is KPMG Peat Marwick's Software Quality Center, based in Boston. John Crawford, Director of the Center, describes the primary goal of a verification test. "We primarily focus on establishing, from a technical perspective, that the escrow deposit is what it purports to be. This is most often accomplished by understanding the terms of the escrow agreement, and then preparing simple and objective tests to verify that the software escrow deposit contains all of the components contemplated by the parties when they entered into the agreement."

Such tests, usually performed at the point the initial deposit is made, are able to provide detailed inventories of documents, directories of tapes and analysis of electronic materials. The testing plan is

designed following discussions between the testing agent, a programmer representing the developer and a technical representative of the licensee. These verifications often include a combination of the following goals:

- (1) Verify that the appropriate software modules and supporting documentation are present.
- (2) Verify the technical integrity of the materials through the successful execution of a compilation.
- (3) Verify that the assembled system performs its intended function by executing a sample of system transactions or usage scenarios.
- (4) Independently collecting all escrow materials and securing them in the escrow deposit.

Once a verified version is secured, it should not be removed from the escrow deposit unless an updated version is tested. Through these comprehensive tests, the licensee will have the assurance they seek that the deposit is complete and reliable. The most beneficial test will compile the software using the escrow materials and test the yielded product. If any deficiencies are discovered, the escrow agent will report the test results to the licensee and work with the developer to upgrade the deposit. The cost for these services are subject to the complexity of the software product and the test design. Such fees are quoted on a per project or per hour basis and usually range from \$150 to \$250 per hour.

Filing For a Release

Most professional escrow agents report approximately five percent of escrow accounts are released to a beneficiary. Yet this possibility must be considered by any company entering into this arrangement. Most requests for a release are initiated by the licensee because the developer has either ceased operations or failed to support the product. In 1988, the United States Bankruptcy Code (Sec. § 365

N) was modified to provide protection to companies licensing technology. This step excluded escrow agreements from the protection awarded to a technology company when it files for bankruptcy. To initiate most releases, the licensee contacts the agent and provides any documentation that is required by the escrow contract to support the request. The agent then notifies the developer and the developer is given a period of time to object or consent to the request. More often than not, the developer will rectify any problem with its licensee during the period following the request for release. However, the developer may contest the release of the materials and take the matter before a forum designed to resolve such disputes, such as a court or arbitration panel. During this process, the escrow agent should be expected to be responsive and encourage communication between all parties, including counsel. If any questions arise, all parties should look to the agreement for direction.

There is a less common procedure for a release that is referred to by many as a "quick release." This condition is based upon a release process which allows a licensee to request the deposit and immediately receive the materials from the agent. The developer has no power to stop the release, but can appeal to a court or arbitrator to reverse the action. While the developer may eventually reverse the process, the licensee is able to use the materials to support its operations. Most developers object to this arrangement on the grounds that it does not protect their interest in the process and does not guard against unjustified release requests. However, this scenario does allow the licensee to avoid any lengthy delays in support that ensue during a protracted battle between the licensee and developer. One set of conditions that may address this issue is "restrictions on use of the escrow materials."

Restrictions on use of the deposit materials are often placed in the escrow contract and are not necessarily tied to a "quick release". These terms are often similar, if not identical, to confidentiality terms that are found in the license agreement. These restrictions will identify how the materials may be used, such as

limiting their use to only supporting the existing product and not allowing major modifications. Other topics, such as who may have access to the materials within the Licensee's company or at what site the materials must remain, are also addressed. These conditions must reach a balance between the protection of the developer's interest and the ability of the licensee to use the escrow materials effectively if they are released.

Conclusion

The escrow industry is similar to the technology industry from which it was born - it is constantly changing its services in an effort to improve itself. Admittedly, some escrow arrangements fail to benefit a technology licensee. However, to disregard escrow services ignores one of the few options that are open to a licensee to secure long-term support for their licensed programs. There is nothing inherent in the escrow process that prevents it from protecting a the company that invests the proper time to the project. The escrow must be analyzed by all involved parties and established under the premise that it will eventually be used. The strong escrow contract forms the necessary foundation to the service. When a professional, impartial agent is employed, the service will provide both assurance and potential benefits to the involved parties.

¹ The Economist, February 18, 1995, pages 63-63.

² The Law Works, April 1995, page 13

The Advanced Intelligent Network—A Security Opportunity

Thomas A. Casey, Jr.

GTE Government Systems Corporation
Communication Systems Division

77 A Street
Needham, MA 02194

617-455-4075

caseyt@mail.ndhm.gtegsc.com

Abstract

The public switched telephone network (PSTN) is evolving from a closed network made up of specialized equipment into an open network employing many of the same components and protocols that are used in the Internet. The security vulnerabilities of the Internet are well known. The possible introduction of these vulnerabilities into the PSTN provides opportunities—for hackers to exploit the vulnerabilities and for security professionals to eliminate them.

The current PSTN is evolving into what is known as the Advanced Intelligent Network (AIN). In the old PSTN, the control functions for telephone services (service logic) are implemented in software that runs in telephone switches. In the AIN, service logic is implemented by Service Logic Programs (SLPs) that run in Service Control Points (SCPs). SCPs are, in most cases, ordinary commercially available microprocessor-based workstations or servers, running the same insecure operating systems that are used on most Internet hosts. SCPs communicate with switches through the SS7 network. In addition, SCPs will have connections (sometimes via other machines) to the telephone companies' corporate data networks to support such functions as customer service and billing. There are also plans to offer customers an Internet interface for changing their service parameters—such as the number to which their calls should be forwarded.

The obvious and very interesting potential security problems created by these changes in the PSTN have received comparatively little attention from the information security community. It is the objective of this paper to change that.

1. Introduction

The public switched telephone network (PSTN) is currently undergoing some radical changes. In the past, it was a closed network made up of specialized equipment that very few people understood. Connection of customer equipment to the voice network was strictly regulated, and the control system was completely closed. Over the years, many restrictions on the connection of customer equipment to the voice network have been eliminated. Currently the same thing is happening to the control network—the SS7 network. It is evolving into an open network employing many of the same components and protocols that are used in the Internet. Connection of third-party equipment to the SS7 network is being mandated, both by federal regulations and by the marketplace. It appears that there will eventually be connections between the SS7 network and the Internet. The possible introduction of the well-known Internet security vulnerabilities into the PSTN provides opportunities—both for hackers to exploit the vulnerabilities and for security professionals to eliminate them.

In North America, Advanced Intelligent Network (AIN) is the term for the changes that the PSTN is undergoing. (In the rest of the world, these changes are known as the Intelligent

Network (IN), because of differences in the evolutionary path taken by the PSTN in various parts of the world.) Section 2 of this paper is a summary of AIN concepts and terminology.

The essence of the AIN is this: In the old PSTN, the control functions for telephone services (service logic) are implemented in software that runs in telephone switches. Implementation of new services requires the (costly and risky) modification of software in thousands of switches, of a variety of models and ages. In the AIN, service logic is implemented by Service Logic Programs (SLPs) that run in Service Control Points (SCPs). New services can be implemented in one SCP (or in several, for reliability and performance reasons). One SCP can serve many switches, communicating with them via the SS7 network.

The factors contributing to the AIN security problem include the following: SCPs (and the other new components introduced by the AIN) are, in most cases, ordinary commercially available workstations or servers, running the same insecure operating systems that are used on most Internet hosts. Service logic programs are, in many cases, ordinary application programs developed by persons without any particular expertise in security. Further, the AIN objectives of rapid development and deployment of new services in response to changes in the marketplace tend to be in conflict with objectives of software correctness for reliability and security. SCPs will have connections (sometimes via other machines) to the telephone companies' corporate data networks to support such functions as customer service and billing; some of these networks are connected to the Internet. There are also plans to offer customers direct Internet interfaces for changing their service parameters—such as the number to which their calls should be forwarded. Such user interfaces could place the integrity of customers' service parameters at risk, and the Internet connections supporting them could place the integrity of the entire network at risk.

The AIN brings with it a number of interesting and challenging security problems. These problems have received relatively little attention, possibly because the inner workings of the PSTN and the AIN are unfamiliar to most members of the information security community. It is the objective of this paper to stimulate interest in these problems within the security community.

2. AIN Concepts and Terminology

This section is a very much oversimplified discussion of the AIN. It includes a high-level summary of the AIN architecture, the functions of the major components, and definitions of some of the acronyms. It is provided here in hopes of helping the reader unfamiliar with the AIN to make some sense of the terminology and the multitude of acronyms. The indulgence of readers familiar with the AIN is requested. More complete information about the AIN can be found in [Robrock91] and the extensive list of references in it.

Figure 1, on the next page, shows the major AIN components and their relationships to one another. A small subset of the total network is shown, containing at least one example of each AIN component and of the network connections between them.

In this figure, the thin solid lines represent signaling links; these links carry messages associated with the setup and teardown of individual calls. The majority of the signaling links are part of the SS7 network; the signaling links connecting the SSP to the ADJ and IP are exceptions. The thick solid lines represent voice links. Notice that the ISDN link contains both a voice and a signaling channel. The OAMP (Operation, Administration, Maintenance, and Provisioning) links carry messages associated with service deployment, with the provisioning (initial setup) of services for individual customers, and with updates to customers' service specifications. The OAMP links employ various protocols, including TCP/IP and X.25.

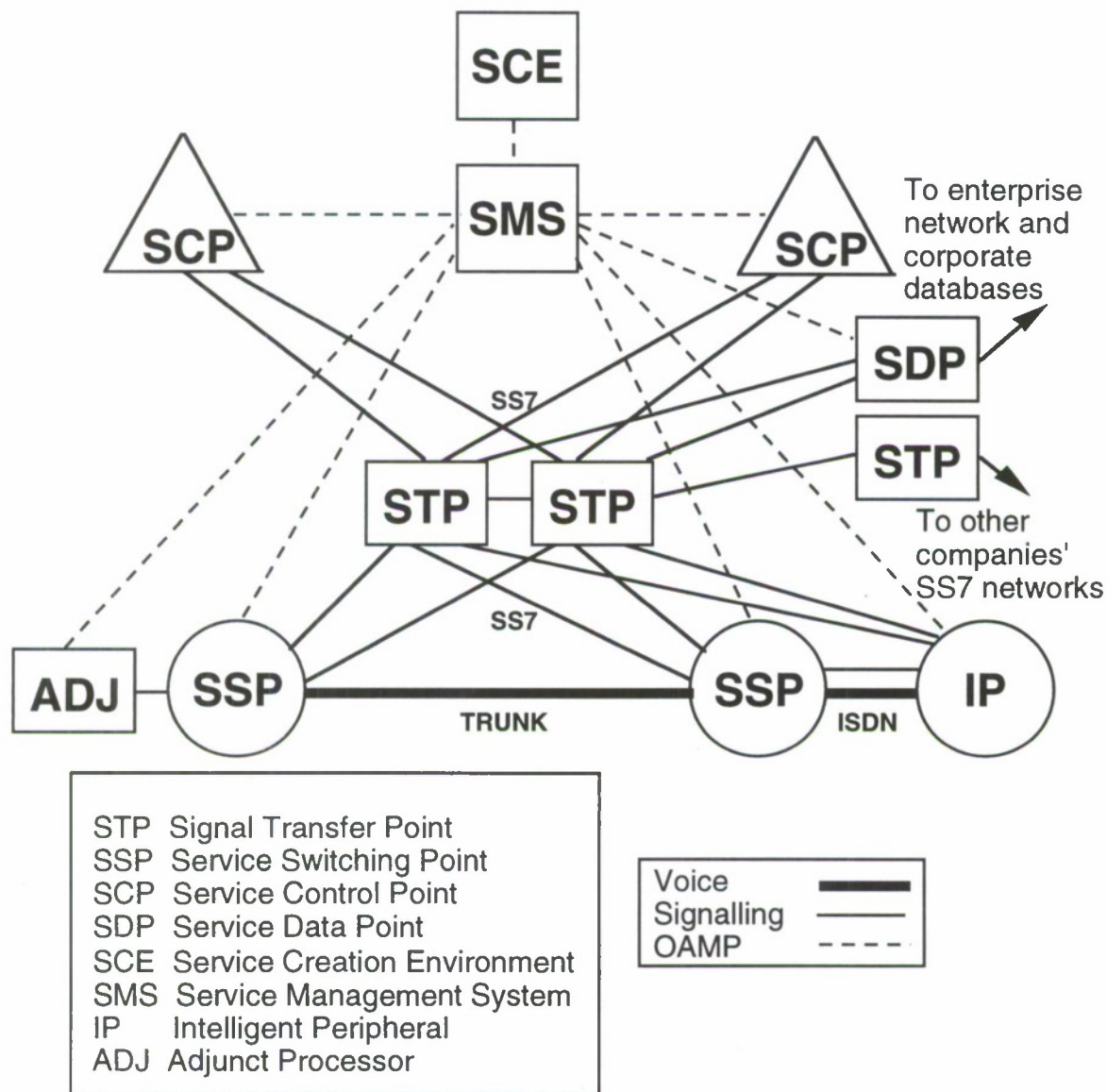


Figure 1 - AIN Architecture

2.1 AIN Terminology

AIN has been said to stand for "Acronym-Intensive Network." Most of the AIN components have three-letter acronyms beginning with S and ending with P or F. Most of the S's stand for "Service," most of the P's stand for "Point," and the F's stand for "Function." A point is the computer on which an architectural function is implemented, and "point code" is the term for an SS7 network address.

For each AIN component in Figure 1, Table 1 gives the acronym, the words that it stands for, and a very brief description of the component. The text that follows Table 1 briefly outlines the evolution of the AIN and describes each of the components in more detail.

SSP	Service Switching Point	Telephone switch, e.g., 5ESS, GTD-5
STP	Signal Transfer Point	SS7 packet switch
SCP	Service Control Point	The brain of the AIN; runs Service Logic Programs (SLPs)
SCE	Service Creation Environment	Development environment for SLPs
SMS	Service Management System	Deploys SLPs; provisions services for customers; updates service records
SDP	Service Data Point	Database Server for SCPs
IP	Intelligent Peripheral	Recorded messages, voice response, collection of PINs, some service logic
ADJ	Adjunct Processor	Provides SCP-like services to directly-connected switch(es) (SSPs)

Table 1 - AIN Components

2.2 Evolution of the AIN

In earlier switching systems, call setup signals were sent over the trunk lines between switches using tones similar to those emitted by touch tone phones. In Figure 1, these switches are labeled SSP (Service Switching Point).

Hackers discovered that they could build devices which they called blue boxes. These blue boxes could imitate the call setup signals and set up calls while bypassing the accounting for the calls. Thus, they were able to steal long-distance phone service.

Common Channel Signaling (CCS) eliminated this security flaw. Call setup signals are now sent between switches using a packet-switched network. The packet switches are called Signal Transfer Points (STP). The latest version of the CCS system is SS7 (Signaling System 7). At a very high level, there is some resemblance between the SS7 network and a TCP/IP network such as the Internet. However, at a more detailed level, they are quite different. The SS7 network is optimized and specialized to provide the highly reliable, real-time transfer of telephone call setup signals. More information on the SS7 network can be found in [Modaressi90].

The next step introduced a certain class of services, examples of which are nationwide 800 numbers and nationwide calling card services. These services are implemented by service logic in every switch. This logic queries either an 800 number translation database or a calling card database, in a database server known as a Service Control Point (SCP). Communication between the switch and the SCP is provided by the SS7 network. These services are, as we know, available today. (The reader might notice a discrepancy between this description and the terminology in Table 1, which gave the name Service Data Point (SDP) to the database server. The table gives the modern terminology. If we were to describe the earlier architecture using the modern terminology, we would say that the Service Data Function (SDF) was implemented in the SCP rather than in a separate SDP.)

The latest step, which the industry is now in the process of taking, moves service logic out of the switches and into the SCPs. Switches will now have trigger points in the call setup logic. At these points they can (if the appropriate trigger is enabled) send a query to an SCP asking how to proceed in the call setup. An SCP can implement one, or several, AIN services. The logic for a new service need not be added to every switch (SSP); rather, it can be implemented in one SCP (or in several for performance and availability reasons). Once the call setup logic in all switches is upgraded to include the triggers, new services can be created without requiring further modifications to switches.

2.3 AIN Components

The Service Switching Point (SSP) is a telephone switch. SSPs are present in the existing, pre-AIN PSTN. In order to participate in the AIN, a switch must be upgraded to run a version of software that conforms to the AIN call model and has triggers at specified points in the call setup sequence. If a trigger is enabled, the SSP will, at that point in call setup, send a request to the SCP asking for instructions about how to proceed with the call setup. Triggers can be enabled or disabled selectively, for individual lines, groups of lines, or the entire switch.

The Signal Transfer Point (STP) is an SS7 packet switch. These, too, are part of the existing network. There are few, if any, high level architectural changes required to the STP to support AIN services, although some detailed changes are probably required. It is likely that significant changes would be required to support enhanced security.

The Service Control Point (SCP) is the brain of the AIN. It runs Service Logic Programs (SLPs), which control call processing and provide all the new AIN services. The switch (SSP) will consult the SCP at various points in the call setup sequence. The SCP will run its Service Logic Programs, consult its (customer-specific) databases, and return instructions to the switch. There is a requirement that the instructions be returned very quickly since the switch is in the middle of a call setup and the customer is waiting for the ringing tone to start. An SCP can provide service to multiple switches. The switch and SCP communicate over the SS7 network.

The Service Data Point (SDP) is a database server for the SCPs. It implements the Service Data Function (SDF). It contains the customer-specific databases that are queried by SLPs during call setup. In earlier versions of the network, the SDF was implemented in the SCP (that is, the SCP contained its own databases). A separate database server (the SDP) is more desirable for practical reasons: there is sometimes a requirement that several SCPs be able to query a single database, and a combined SCP-SDP, along with its surrounding network, could become overloaded.

The Intelligent Peripheral (IP) serves a switch (or perhaps several switches), to which it is connected by an ISDN link. It provides such services as recorded announcements, voice recognition, and the collection of DTMF tones for later transmittal, when a customer, for example, is entering a PIN number. The Adjunct Processor (ADJ) provides the adjacent SSP (to which it is connected by an Ethernet link) with SCP-like services requiring faster response than can be obtained over the SS7 network from remote SCPs. Both the ADJ and the IP can run some SLPs.

The Service Creation Environment (SCE) is a development environment for Service Logic Programs (SLPs). The Service Management System (SMS) provides an interface between the SCE and the SCP for deploying new SLPs. It also provides other management functions such as the provisioning (initial setup) of services for customers, and the updating of individual customers' call processing options.

The OAMP network is separate from the SS7 network. It connects the AIN components and carries traffic not related to the setup of telephone calls. In particular, it carries traffic related to

Operation, Administration, Maintenance, and Provisioning. (Provisioning is the term used to describe the initial setup of a new service for an individual customer, as opposed to either the installation of a new service network wide, or the changing of service parameters by an individual customer on a service already provisioned.) The OAMP network uses standard protocols such as TCP/IP and X.25. It is connected to other corporate networks, and it may be connected to the Internet by gateways or dual-homed hosts.

The SSP and STP, along with the pre-AIN SCP, are part of the existing network. The new SCP, as well as the SCE, SMS, SDP, IP, and ADJ, are all being added to the network as part of the AIN. They are being implemented, for the most part, with the same standard, commercial, insecure workstations, servers, and operating systems that are found on the Internet. The AIN services are provided by ordinary application programs, written by people without any special security expertise. It is an objective of the AIN architecture to allow new telephone services to be created and deployed rapidly and inexpensively.

2.4 AIN Standards

The AIN architecture and protocols are continually being defined and refined by national and international standards groups. In North America, the work is being carried out by Standards Committee T1 - Telecommunications, Technical Subcommittee T1S1. In the international arena, the work is being done by the International Telecommunications Union (ITU), Telecommunication Standardization Sector.

Most of the work of these standards bodies has been focused on providing functionality and interoperability. In recent years, they have shown some interest in adding security to the AIN. However, while a standards body may be the appropriate forum in which to choose among several developed and tested communication protocols for worldwide standardization, it is, perhaps, a less effective forum in which to design and debug new and creative solutions to difficult security problems. In the author's opinion, it would be better for these solutions to be developed, and at least prototyped, by telephone companies and equipment vendors before being standardized.

3. The Security Problem

The magnitude of the potential security problem in the worldwide telephone network is so great that it is difficult to describe. There are potential security problems in almost all parts of the network. This paper attempts to provide a comprehensive outline of the threats and vulnerabilities, and to give a few examples. However, space limitations, as well as reluctance to describe vulnerabilities in great detail, have made it necessary to leave it to the reader's security background, expertise, and imagination to supply many of the details.

The old network was secured (to the extent that it was secured) mainly by obscurity, isolation, and physical barriers. The equipment and protocols were understood by few people, and the network interfaces were few and of limited functional capability. The AIN changes all that, by adding a great deal of new network components, protocols, connections, and interfaces.

The new components and protocols are, in many cases, identical to those that are used on the Internet; their vulnerabilities are well known. The AIN adds a great deal of new network connectivity: SS7 connections to the new components, new OAMP connections between all components (using protocols such as TCP/IP and X.25), and connections to other networks, including, in some cases, the Internet. The new customer interfaces range from the now familiar touch tone interface ("press 1 if ..."), to Internet (World Wide Web) interfaces allowing customers to change their service parameters, to direct connection of commercial customers' computers to the SS7 network. In addition to customer interfaces, there are new interfaces for

telephone company employees engaged in business functions, customer care functions, and network maintenance functions. There are also new interfaces for AIN service developers (e.g., SCP writers) who may not be telephone company employees.

The new components will add vulnerabilities to the PSTN. The new connections and interfaces will make the network more accessible to those who would try to exploit those vulnerabilities. The new services, which customers will come to depend on for the conduct of their business and personal lives, will make the PSTN an even more attractive target for hackers, unscrupulous insiders, and those who would hire them in an attempt to gain some advantage for themselves or to place the telephone company or its customers at some disadvantage.

3.1 Vulnerability of Interfaces

Vulnerabilities are security weaknesses in the network. In considering vulnerabilities, we should look at all the interfaces through which an intruder might attack the telephone network. We will discuss the following interfaces:

- Customer interface
- Telco business interface
- Telco maintenance interface
- Service creation (application program) interface
- Lower layers, where there wasn't supposed to be an interface
- Law enforcement interface

The customer interface consists of the familiar touch-tone phone, plus all the other interface elements such as PBXs, pagers, cellular phones, personal computers, and in the near future, the Internet. Using this interface, customers can place calls (and by implication, agree to pay for them). They can also change their service parameters. In both cases, good security practice would require that customers be authenticated. The authentication mechanisms must be reliable, but they must also be convenient enough that customers are willing to use them. Currently popular mechanisms, such as passwords or PINs (4-digit numbers) are of dubious reliability.

Examples of abuse of the current customer interface include shoulder surfing (stealing a calling card number by watching someone make a call at a public phone), and cellular cloning (stealing the ESN (serial number) and MIN (phone number) off the air and using them in a clone phone). These abuses cost the telephone industry a great deal of money and cause customers a great deal of annoyance. The AIN adds more user interfaces and gives the users (legitimate or otherwise) of those interfaces more power to control the customer's services. For example, a call forwarding service could be abused to steal customers from a competitor (see [NYT95]).

The telco business interface is used by such people as customer service representatives, as well as people carrying out billing functions and the like. These people have access to see and modify information belonging to many customers. Because of its power, this interface should be protected by stronger authentication than that used for individual customers, to protect against the outsider threat. In addition, this interface should have its power limited by least privilege considerations, and its use should be audited, to protect against the insider threat. It is intended that the AIN will add a great deal of new and potentially complex services as quickly as the marketplace is ready for them. The number and complexity of the new services will require that customer care personnel have the ability to fix problems rapidly. If the interfaces provided to allow this were to be abused by a clever outsider or an unscrupulous insider, it could cause a great deal of trouble for both customers and the telephone companies.

The telco maintenance interface is used by people whose job it is to keep the network running. This interface includes both centralized network management systems and the direct interfaces to equipment located in, and accessible only through physical access to, switching offices. These interfaces clearly provide the ability to do significant damage to hardware or software, and they must be protected by strong authentication and strong physical security. The National Communications System AIN Program Office has studied the AIN security problem in general. Their report, now out of print, gave particular attention to the vulnerabilities of this interface.

The service creation interface adds a whole new set of vulnerabilities to the network. It is anticipated that new AIN services will be created rapidly, mostly by people who do not work for telephone companies, and who may not have a strong appreciation of the need for reliability and security in the network. A problem currently receiving much attention is the feature interaction problem, in which two services, created independently, can accidentally and innocently interfere with each other's operation. The existence of this problem points up the fact that the AIN architecture contains no provisions to prevent such interference, whether it be accidental or deliberate. The security implications of Trojan horses and trapdoors in SLPs are obvious, as are the implications of exploitable bugs introduced by accident.

The lower layer interface, where there wasn't supposed to be an interface, is included to remind the reader that one of the ways that hackers break into systems is by finding, or inventing, interfaces that were not supposed to exist. These interfaces will not be found in any design document. They exist, or the potential for them exists, only as an accidental byproduct of implementation details. (The sorts of things we have in mind here include putting a monitor on a LAN, or sending a long message that overflows a buffer and provokes the receiving program into incorrect, and possibly insecure, behavior.) At a recent telephone industry trade show, one vendor was displaying an SS7 network monitor that could passively intercept, log, and interpret all SS7 traffic.

The law enforcement interface offers a multitude of vulnerabilities. This interface is legally mandated by the Communications Assistance for Law Enforcement Act (CALEA). It requires service providers to enable the execution of lawfully authorized wiretaps and call traces. In the future this process will become more automated than in the past, allowing intercepts to be carried out by remote control. In the absence of strong security measures, the possibilities for abuse of this facility are obvious. They include: unlawful intercepts for purposes of blackmail, overzealous law enforcement, or interference with legitimate law enforcement (possibly endangering law enforcement personnel); release of lawful intercept data to unauthorized persons; suppression or alteration of intercept data to protect the guilty; and falsification of intercept data to incriminate the innocent.

3.2 Threats to the Network

Threats are actions that an intruder might take to attack the network by exploiting its vulnerabilities. The threats to the PSTN are too numerous to mention individually. This section only outlines threats and attack methods. It is best read slowly, using one's imagination.

Threats can be placed in four categories: theft of information, unauthorized alteration of information, denial of service, and theft of service. These threats can be carried out using a variety of attack methods.

The network could be attacked by three methods: physical access to network nodes or links, network access to network nodes, or the introduction of malicious software during the software development or software distribution processes. In addition, individual applications could be attacked at the end user interface by attempting to exploit weaknesses in their user authentication and usage authorization features, or by probing for flaws in their handling of incorrect input.

Attacks based on physical access to nodes could be carried out by insiders abusing their authorized access to nodes, by employees abusing their building access to gain unauthorized access to nodes, or by intruders who breach building security. Having gained access to a node, an intruder or insider could alter hardware or software, or make use of maintenance interfaces. It is possible to steal end-user or network control information, alter both types of information, or sabotage the node. Having access to, and unlimited control over, a node, an intruder could use it to launch network-based attacks on other nodes.

Network-based attacks could come from a compromised node in a telephone company's own SS7 network. Also, with the advent of mediated access, they could come from the networks of other telephone companies or third party service providers, due either to unscrupulous insiders or to lax physical security that allows intruders to gain access to nodes. In addition, an intruder having physical access to a link could attach computing equipment to it and use that equipment to carry out network-based attacks.

There are two categories of network-based attacks: passive and active. Passive attacks involve the monitoring of messages and the theft of end-user or network control information. Active attacks involve the sending of messages, often with forged sender IDs. These messages are calculated to induce the receiver to take some improper action that will result in a successful attack in one of the four threat areas (theft or alteration of information, denial of service, or theft of service). Often such messages exploit known bugs in the software in the receiving node. Defenses against both categories of network based attack involve the use of cryptography. It can provide message privacy, message authentication/integrity, or both.

Unscrupulous software developers will sometimes insert Trojan horses or trapdoors into their programs. These are pieces of malicious code that will carry out some covert function when they are installed in a production system, possibly including allowing the author to break into the system, bypassing its security features. Malicious code could also be inserted during distribution of software to the network nodes. It is an objective of the AIN architecture to allow SLPs to be written quickly and easily by a diverse set of individuals and organizations. Potentially, new SLPs could become part of the body of operating PSTN software with little or no control being exercised over their quality, correctness, or freedom from malicious code. The SLP execution environment does not impose any least-privilege constraints. Apart from implementation difficulties (involving conflicting objectives of assurance and efficiency), it is not clear how such constraints could be specified without interfering with AIN functionality. One possible solution—rigorous inspection of SLPs—is in conflict with the objective of rapid service deployment. The correct solution to this problem is not obvious.

3.3 Consequences of Attacks

Successful attacks in any of the threat areas discussed above could allow the perpetrator to accomplish one or more of a large number of consequences that are beneficial to the perpetrator but harmful to the owners and legitimate users of the network. These consequences could result in damage to individual customers—both residential and commercial—and to telephone companies. In some cases they could even threaten public safety, the national economy, or the national security. As was the case with threats, the consequences are too numerous to mention individually. They fall into the following general areas.

- Theft of private end-user information, such as voice conversations, voice mail, or data
- Theft of private telephone company information, such as customer lists, calling card numbers, or cellular authentication codes
- Alteration of end-user or telephone company information for the purpose of damaging the information resources of the victim

- Theft of, or alteration of, network control information to facilitate further penetration of the network
- Selective interference with the services of certain individuals or firms, for purposes of harassment or unscrupulous competition
- Widespread interference with network services (i.e., sabotage), or the threat of it, for purposes of terrorism or extortion
- Theft of telephone services

4. Conclusions

The problem described above is a large, multi-faceted information system security problem. It involves both computer security and network security. The problems exist in all layers, from the lowest layers of network infrastructure, up through the execution environment of application software, up to the design of the end-user interfaces.

Many of these problems could be solved by the proper application of existing computer security and network security technology. Encryption, for message privacy, message authentication, and message integrity, could provide defenses against many of the network based attacks. State of the art user authentication methods, such as smart cards for customers and telco employees, and biometric devices (e.g., fingerprint readers) controlling physical access to buildings and rooms housing switching equipment, would provide good defenses against attacks based on physical access or user interface exploitation. High assurance operating systems (those having Orange Book ratings of B2 and above) would be free of many of the exploitable vulnerabilities in the non-rated systems currently being used for AIN components. High assurance operating systems are expensive, but quantity discounts might be available if they were to be purchased in the numbers needed for the entire PSTN.

Collectively, the attendees of this conference probably have the necessary security expertise to solve the PSTN security problems. However, to secure a system it is necessary not only to understand security but also to understand the system being secured. In the author's experience, people having both security expertise and a thorough understanding of the PSTN and the AIN are rare. If these problems are to be solved, security experts and PSTN experts will have to work together and educate each other in their respective areas of expertise.

The communications network that we all depend on is undergoing changes that make it increasingly vulnerable to security problems. We, as security professionals, should work to bring these problems to the attention of the appropriate decision-makers in the telephone industry, and should take a personal interest in helping to solve them.

References

- [Modaressi90] Modaressi, A.R, and R.A. Skoog, "Signaling System No. 7: A Tutorial", *IEEE Communications Magazine*, July 1990, pp. 19-34.
- [NYT95] "Plumber Is Charged In Call Forwarding Theft", New York Times NATIONAL, Sunday, January 29, 1995, p. 32.
- [Robrock91] Robrock, R.B., "The Intelligent Network—Changing the Face of Telecommunications", *Proceedings of the IEEE*, Vol. 79, No. 1, January 1991, pp. 7-20.

**THE ADVANCED INTELLIGENT NETWORK
A SECURITY OPPORTUNITY**

19th National Information Systems Security Conference
October 1996

Thomas A. Casey, Jr
GTE Government Systems Corporation
Communication Systems Division

617-455-4075
caseyt@mail.ndhm.gtepsc.com

**THE ADVANCED INTELLIGENT NETWORK
A SECURITY OPPORTUNITY**

19th National Information Systems Security Conference
October 1996

Thomas A. Casey, Jr
GTE Government Systems Corporation
Communication Systems Division

617-455-4075
caseyt@mail.ndhm.gtepsc.com

**THE ADVANCED INTELLIGENT NETWORK
A SECURITY OPPORTUNITY**

19th National Information Systems Security Conference
October 1996

Thomas A. Casey, Jr
GTE Government Systems Corporation
Communication Systems Division

617-455-4075
caseyt@mail.ndhm.gtepsc.com

**THE ADVANCED INTELLIGENT NETWORK
A SECURITY OPPORTUNITY**

19th National Information Systems Security Conference
October 1996

Thomas A. Casey, Jr
GTE Government Systems Corporation
Communication Systems Division

617-455-4075
caseyt@mail.ndhm.gtepsc.com

The Opportunity

- **Public Switched Telephone Network (PSTN) is becoming the ...**
- **Advanced Intelligent Network (AIN)**
- **New and wonderful telephone services**
- **New network components**
- **New vulnerabilities**
- **New opportunities for security professionals**

- ## The Opportunity
- **Public Switched Telephone Network (PSTN) is becoming the ...**
 - **Advanced Intelligent Network (AIN)**
 - **New and wonderful telephone services**
 - **New network components**
 - **New vulnerabilities**
 - **New opportunities for security professionals**

The diagram illustrates the AIN (Advanced Intelligent Network) Architecture. It shows a central STP (Signal Transfer Point) connected to various other components. The components are arranged in a hierarchical and interconnected manner. The top row includes SCE (Service Creation Environment), SMS (Service Management System), and SCP (Service Control Point). Below these are SDP (Service Data Point) and STP (Signal Transfer Point). The bottom row includes ADJ (Adjacent Processor), SSP (Service Switching Point), and IP (Intelligent Peripheral). The diagram uses different line styles to represent different types of connections: solid lines for Voice, dashed lines for Signalling, and dotted lines for OAM (Operation, Administration, and Maintenance). The connections are as follows: SCE is connected to SMS and SCP. SMS is connected to SCP and SDP. SCP is connected to SDP and STP. SDP is connected to STP. STP is connected to ADJ, SSP, and IP. ADJ is connected to SSP. SSP is connected to IP. The diagram also shows connections between the STP and the other components, indicating a central role for the STP in the network.

AIN Architecture

Legend:

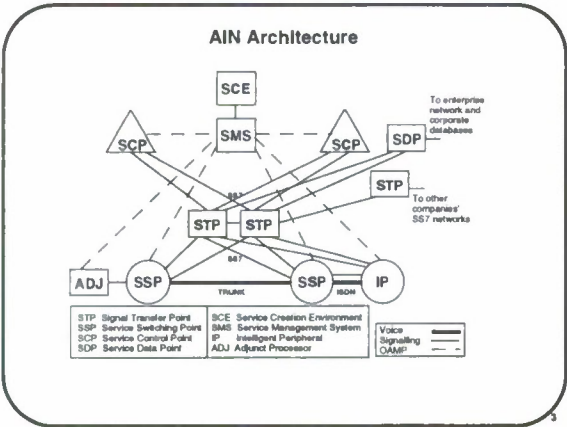
- Voice
- - - Signalling
- ... OAM

Components and their connections:

- SCE** (Service Creation Environment) is connected to **SMS** (Service Management System) and **SCP** (Service Control Point).
- SMS** is connected to **SCP** and **SDP** (Service Data Point).
- SCP** is connected to **SDP** and **STP** (Signal Transfer Point).
- SDP** is connected to **STP**.
- STP** is connected to **ADJ** (Adjacent Processor), **SSP** (Service Switching Point), and **IP** (Intelligent Peripheral).
- ADJ** is connected to **SSP**.
- SSP** is connected to **IP**.

Connections between components:

- SCE** to **SMS**: Solid line (Voice)
- SCE** to **SCP**: Solid line (Voice)
- SMS** to **SCP**: Dashed line (Signalling)
- SMS** to **SDP**: Dashed line (Signalling)
- SCP** to **SDP**: Dashed line (Signalling)
- SCP** to **STP**: Dashed line (Signalling)
- SDP** to **STP**: Dashed line (Signalling)
- STP** to **ADJ**: Dashed line (Signalling)
- STP** to **SSP**: Dashed line (Signalling)
- STP** to **IP**: Dashed line (Signalling)
- ADJ** to **SSP**: Solid line (Voice)
- SSP** to **IP**: Solid line (Voice)



AIN Components

- **Current**
 - **SSP: telephone switch**
 - **STP: SS7 packet switch**
- **New**
 - **SCP: service control point**
 - **SCE: service creation environment**
 - **SMS: service management system**
 - **SDP: service data point**
 - **IP: Intelligent peripheral**
 - **ADJ: adjunct processor**

- ## AIN Components
- **Current**
 - **SSP: telephone switch**
 - **STP: SS7 packet switch**
 - **New**
 - **SCP: service control point**
 - **SCE: service creation environment**
 - **SMS: service management system**
 - **SDP: service data point**
 - **IP: Intelligent peripheral**
 - **ADJ: adjunct processor**

Recent Incidents

- **The Insider**
 - stole 140,000 calling card numbers
 - \$140 million worth of fraudulent calls
- **The Hacker**
 - gained control of central offices
 - made a home telephone ask for money
 - cut off parole officer's phone service
 - listened to FBI calls
- **The Plumber**
 - stole customers from competitors
 - had their calls forwarded to his office

- ### Recent Incidents
- **The Insider**
 - stole 140,000 calling card numbers
 - \$140 million worth of fraudulent calls
 - **The Hacker**
 - gained control of central offices
 - made a home telephone ask for money
 - cut off parole officer's phone service
 - listened to FBI calls
 - **The Plumber**
 - stole customers from competitors
 - had their calls forwarded to his office

Threat Categories

- **Four major threat categories:**
 - **Theft of Information**
 - **Unauthorized alteration of Information**
 - **Denial of service (sabotage)**
 - **Theft of service**
- **In a network, separate attention should be given to threats to:**
 - **end-user data and processing functions**
 - **network control data and processing functions**

- ## Threat Categories
- **Four major threat categories:**
 - **Theft of Information**
 - **Unauthorized alteration of Information**
 - **Denial of service (sabotage)**
 - **Theft of service**
 - **In a network, separate attention should be given to threats to:**
 - **end-user data and processing functions**
 - **network control data and processing functions**

Methods of Attack

- Physical access to network nodes or links
- Network access to network nodes
- Introduction of malicious software
 - during software development
 - during software distribution
- Probe for and user interface weaknesses
 - user authentication
 - handling of incorrect input

Physical Attacks

- Could be carried out by:
 - insiders abusing authorized node access
 - employees abusing building access
 - intruders breaching building security
- Perpetrator could:
 - alter hardware
 - alter software
 - abuse maintenance interfaces
 - steal or alter end-user or network control info
 - launch network-based attacks on other nodes
- Need: resistance to compromised node attacks

Network-Based Attacks

- Could come from:
 - compromised node in company's own network
 - other company's network, via mediated access
 - physical attack on link (attach a computer)
- Two categories: passive and active
 - passive: monitor messages
 - active: forge or alter messages
 - induce receiver to take improper action
 - exploit bugs in AIN application software
 - exploit known vulnerabilities in O.S.

Software Development/Distribution Attacks

- Unscrupulous software developers can leave in their software:
 - Trojan horses
 - Trapsdoors
- Two varieties of malicious software
- Carry out some covert function when installed in the production system
- Allow attackers to succeed, bypassing security features
- Risk: Trojan horses or trapsdoors in SLPs or other AIN component software

Consequences of Attacks

- Theft of private end-user information
- Theft of sensitive telco information
- Alteration of end-user or telco information
- Theft or alteration of network control information
- Selective interference with services
 - Harassment
 - Unscrupulous competition
- Widespread interference with services
 - Terrorism or extortion
- Theft of telephone services

Conclusions

- Apply existing security technology
 - Encryption: privacy, authentication, integrity
 - User authentication
 - High assurance operating systems
- Standardize solutions after testing them
- Combine security and PSTN expertise
- Sell the idea of AIN security
- Opportunities for security professionals
 - Interesting work
 - Help secure the network that we use

SECURITY ISSUES IN EMERGING HIGH SPEED NETWORKS

Vijay Varadharajan*, Panos Katsavos†

June 7, 1996

Abstract

There is a growing interest in the development of broadband services and networks for commercial use in both local area and wide area networks. In particular, connectionless Switched Multimegabit Data Service (SMDS) and connection-oriented Frame Relay based broadband services are beginning to be offered by a number of major operators in the US and Europe. This paper considers the issues that need to be addressed in the design of security services for such high speed networks. First the relevant characteristics of broadband network interfaces are discussed, some of the existing security protocols for TCP/IP and OSI networks are reviewed, and their suitability for providing security in broadband networks assessed. Then the developed arguments are applied to design security services for the connection-oriented Frame Relay networks. An earlier paper [3] considered the development of security services for the connectionless SMDS.

1. Introduction

There is a growing interest in the development of broadband services and networks for commercial use in both local area and wide area networks. The initial stimulus some ten years ago was the development of Asynchronous Transfer Mode (ATM) for use on broadband networks, under the banner of Broadband ISDN (B-ISDN). Recently there is a real pragmatic drive for broadband services, to meet the demand for increased bandwidth for remote sites inter-connection, and for image and high speed data transfer. Broadband activity now has commercial services under a variety of titles, and most of these fall under the umbrella of Fast Packet Switching (FPS). This is a generic term that refers to the switching process being done at a layer which corresponds to layer 2 in the OSI Reference Model. Some of these networking technologies use ATM techniques such as Switched Multimegabit Data Service (SMDS) [2] (can be offered using ATM) and Dual Queued Data Bus (DQDB) [4], and others not such as Frame Relay.

Although it is possible to appreciate the differences between these technologies in terms of the network infrastructure, it is not very clear what each of them has to offer in terms of supporting applications. In particular, with the development of new applications such as networked multi-media, desktop video-conferencing and entertainment services, the need for such broadband services is constantly growing. Also the interconnection of Local Area Networks (LANs) providing high speed information transfer is becoming a strategic necessity for many enterprises to support their growing number of workgroup-based and backbone-type LANs.

There is also a significant change in the nature of network traffic. It is more and more of the form of bursty traffic characterized by an unpredictable demand for bandwidth of several megabytes. The new generation of networking technologies enable interconnection at high-speeds in the range of Mbit/s or even Gbit/s over very wide areas, which effectively moves the bottleneck from networks to end systems. Furthermore, the user is able to access bandwidth on demand and the user is only charged for the bandwidth actually used. As more and more information (audio, image and data) are transferred over

*Prof. Vijay Varadharajan, University of W.Sydney, Australia. Email:vijay@st.nepean.uws.edu.au. Previously, he headed the Distributed Systems Security Group at Hewlett-Packard Labs.UK.

†HP sponsored student,UK.

such high speed networks, security issues are becoming increasingly critical. One may even argue that the success of a high speed technology in the future will be determined not only by its cost effectiveness but also by the level of trust that can be placed on its performance, security and availability.

This paper considers the issues in the design of security services for high speed networks. Section 2 briefly outlines the characteristics of the various broadband networking interfaces that are relevant to this paper. Section 3 first considers the security threats in this environment and the services required; then it describes the background work done and being carried out in the TCP/IP and OSI arena. Section 4 assesses the adequacy of the earlier work in the broadband context, and then considers the placement of security layer within the broadband protocol profiles, and discusses the rationale behind the different choices. Section 5 applies these arguments in the context of connection-oriented Frame Relay networks.

2. Broadband Network Interfaces

A number of options exist for the provision of wide area broadband communication services: leased lines, N-ISDN (Narrowband ISDN), SDH (Synchronous Digital Hierarchy) cross-connect, Frame Relay, FDDI (fibre Digital Data Interface), DQDB, SMDS, B-ISDN, ATM, and SONET (Synchronous Optical Network). These technologies in effect merge the Public Data Networks world and the Voice Circuit-based Networks world together. In doing so, they lead to a new way of modelling communication over the networks in comparison to the OSI model. For instance, the support of circuit-type traffic such as voice, CD quality audio and video traffic is explicitly taken into account in the design of the broadband protocol reference model. Also traditional protocol reference models such as the DoD TCP/IP suite and the OSI model do not have a separate out-of-band signalling path. All network control is carried out by either management entities at the application level with access to the internals of the layers below, or as in-band peer-layer management protocols.

Figure 1 shows some of the protocol profiles of the network interfaces used in broadband communications and their comparison to the OSI model. Although these broadband systems have different models based on multi-protocol stacks, they offer a standard set of services to users : connectionless (CLS), connection-oriented (CO), and isochronous (ISO) services. Note that the functionality of these network interfaces resembles that of the layers 1 and 2 of the OSI model. For instance, SMDS offers connectionless service, and Frame Relay offers connection-oriented service. In fact, in the LAN to LAN market, at present SMDS and Frame Relay are the best known ways of accessing these multi-megabit backbones. FDDI(II) supports both connectionless and isochronous services, but not a connection-oriented one, while DQDB supports the full range. B-ISDN goes further by assigning two different protocol stacks to the isochronous service, namely one for transfer with strict periodicity and another for transfer with guaranteed delivery latency. SDH/SONET interface is equally capable of carrying all different types of traffic. N-ISDN combines circuit-switching with higher special purpose protocol stacks to make provisions for a relatively wide service spectrum including OSI layer 7 teleservices. In this case, there is only a limited support of the connectionless service over the D-channel.

Each of these technologies is claimed to be suitable for a range of applications, and often an application can be equally supported by more than one technology. For instance, both SMDS (CLS) and Frame Relay (CO) can claim to be suitable for interconnection of LANs. The connectionless service supports interactive applications producing bursts of data, with no special timing constraints, to optimize the utilization of network resources. On the other hand, isochronous service addresses circuit-type traffic with strict timing dependencies. The connection-oriented service supports traffic of either data or circuit-type, offering more efficient management of traffic than the connectionless service by allocating resources within the network. In practice, it is not possible to identify all the uses of a multiservice network. However it is clear that such technologies can support not only classic data applications but also applications with real-time transport requirements. Applications make use of these network interfaces by employing an appropriate protocol stack. The synthesis of these protocol stacks is dependent on the nature of the application to be supported; in general, it will be either in the form of a OSI type (or some similar model such as the DoD TCP/IP) or a single adaptation layer.

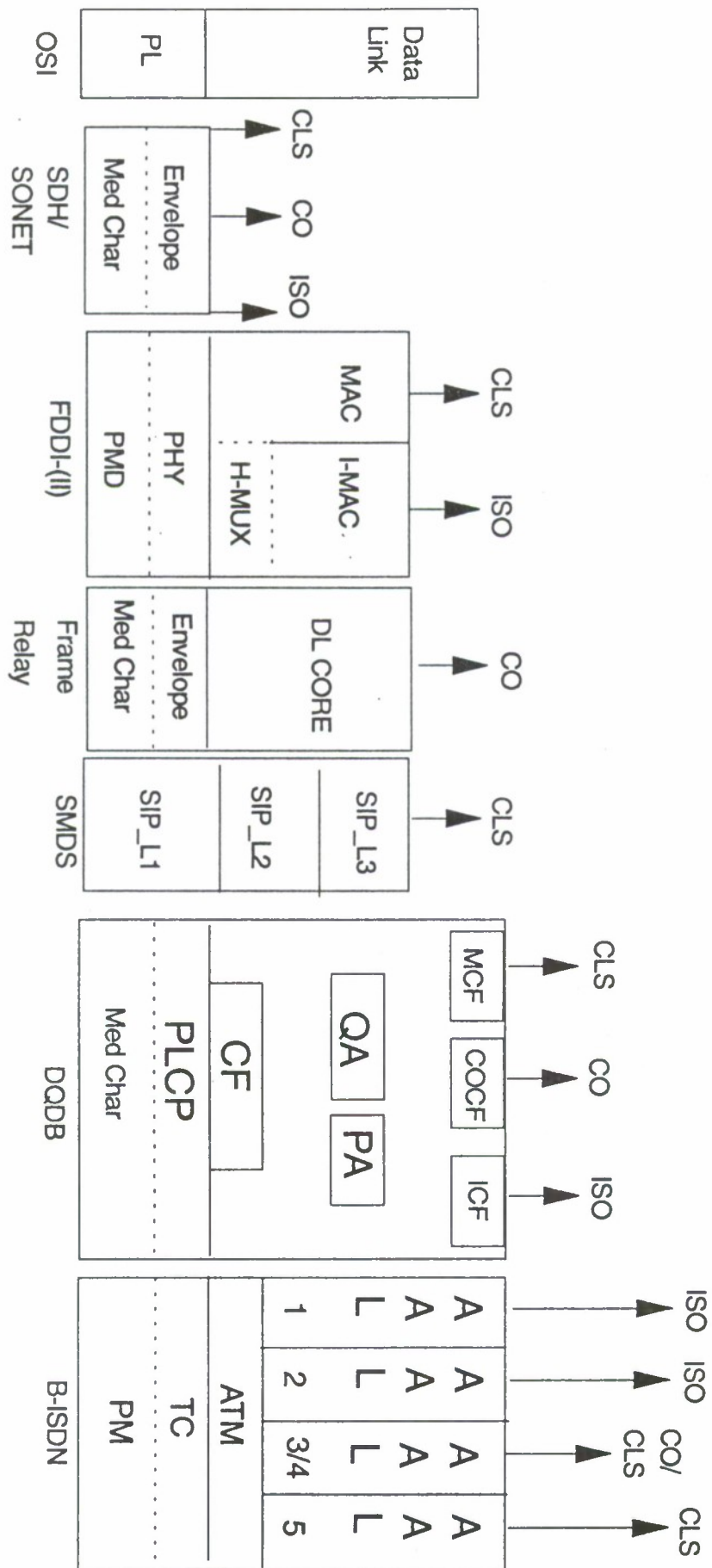


Fig. 1: Protocol Profiles of Broadband Network Interfaces

The end systems continue to be based on conventional multi-layered architectures. The nature of data traffic often demands that the end-to-end transfer be error-free, which is ensured by a robust transport protocol. However implementations of transport protocols such as DoD's TCP, and ISO's TP4 can have performance limitations. Naturally therefore a great deal of research is currently being focussed on this layer; several pieces of work are currently in progress that are considering extensions and modifications to the existing transport protocols to adapt them to high speed environments (e.g. [7]). In our view, it is likely that future communication scenarios will not have full OSI style stack on top of a broadband network interface; some form of adaptation layers will assume the functionality of the traditional transport and network layers.

In the near term, it is clear that most major operators are and will be offering either SMDS or Frame Relay based services. Therefore, it is important at the first instance to address security for these broadband services. Security services for SMDS have been considered in [3]. Before describing the security services for Frame Relay, let us first assess the use of the existing security protocols in traditional networking models to protect broadband information and services efficiently.

3. Security Issues

The fundamental questions that we need to consider when addressing security in high speed Metropolitan Area and Wide Area Networks (MAN/WAN) are :

- What are the security threats in the network environment?
- What are the required security services and mechanisms?
- Where should these services and mechanisms be provided in the protocol stack?
- How are they to be managed?

3.1 Different Organizations' Views

When dealing with security for public networks, the provision of security services need to be considered from different organizations' points of views. There are several options for security services providers within a MAN/WAN environment.

There is the option of the security services and features provided by the *user* to protect his information being carried across public networks. Then there is the provision of security services by the MAN/WAN operator to the end user on a service contract basis. This may not be an attractive option for the end users because of the lack of trust on the network operator. However with the increase in the trend of outsourcing and the options for legislative recourse, there may be opportunities for such a service in the future. In addition to the above, there are security services that are required (and managed) by the network operator to protect his own network resources and information. Finally, there are the services provided by third parties¹ to the end users which may be in the form of supplementary and/or support security services. For instance, notary services come under this category. A common situation is when there is a need to deal with a number of external organizations involving sensitive issues, e.g. contract negotiations. A third party can be of assistance in setting up a secure negotiation between different users through notary and directory services.

From the public network point of view, the positioning of facilities and resources to provide secure traffic needs to be carefully evaluated to ensure that the impact of security features on reliability and overall availability of the services is minimized.

¹Network operators might themselves provide some of these services.

3.2 Security Threats and Services

Let us begin by enumerating briefly the types of security threats that can arise in such a network environment.

Unauthorized disclosure of information via eavesdropping and wiretapping is perhaps the most common threat that comes to one's mind when one thinks about network security attacks. This attack can be carried out by an eavesdropper located anywhere along the communication path. If the target of the eavesdropper is not the user to network interface, subscription to a connectionless service (such as the SMDS, ATM AAL5 or MAC service based on MAN based DQDB) can make the customer traffic less susceptible to this attack compared to the use of a connection-oriented service such as Frame Relay or ATM AAL1/2, where the same route is always followed. A more interesting situation is where the eavesdropper is a legitimate user sharing the network access interface with (or is attached on the same ring as) the source and destination system of the information in transit. This could occur for instance in a multi-CPE (Customer Premise Equipment) access arrangement in an SMDS network.

In addition to this, the information may be altered in an unauthorized manner. The threat of *unauthorized modification* of information and resources causes integrity violation. Such an attack may involve unauthorized insertion and deletion of information transferred over the network. This attack often occurs in conjunction with other attacks such as replay whereby a message or part of a message is repeated intentionally to produce an unauthorized effect. Network parts including digital exchanges, MAN nodes and communication links, as well as bridges, routers and hosts are vulnerable to this form of attack.

In a *masquerading attack* one entity pretends to be another and attempts to gain privileges and access to information and resources to which it is not authorized. For instance, a customer of a MAN can transmit information at a higher rate than the one allowed by the "Access Class" it has negotiated earlier with the network, having at the same time another customer (with whom it shares the user to network interface) to be charged for the bandwidth it uses, as long as the packets it sends carry the other user's source network address (e.g. MAC address in E.164 format).

A noticeable weakness in the general MAN architecture has been that a user connected to a MAN node has access to all the information passing through the node. This raises a fundamental security problem. It imposes limitations on how users can be connected to the network. For instance, when the DQDB protocol is being used on a shared medium access link, each group has to have either a separate access network (as provided by ETSI) or a special interface to the node which prevents access to DQDB slots or be provided with specific security facilities.

Another common attack is the *unauthorized access* to network resources and services. Having successfully masqueraded as another entity, an entity can gain access to resources which are otherwise denied to it. Resources could be network components such as printers or network resources such as operating systems, databases and applications.

Unauthorized denial of service attack by an entity involves the denial of a service to another entity even though the latter is authorized to access that service. That is, an entity prevents other entities from carrying out their legitimate functions. In a network, this form of attack may involve blocking the access to the network by continuous deletion or generation of messages so that the target is either depleted or saturated with meaningless messages. Network failures and errors resulting from equipment reliability also need to be accounted for. For example, an ATM switch may suffer from an accidental breakdown or malfunction resulting in disruption of its customers communications. Denial of a service can be regarded as an extreme case of information modification in which the information transfer is either blocked or drastically delayed.

Repudiation of actions is another form of attack that can occur in a networked system. It occurs when a sender (or a receiver) of a message denies having sent (or received) the information.

3.3 Background

There has been a number of efforts in the development of security protocols for the TCP/IP suite and the OSI over the last years.

3.3.1 TCP/IP Security

Originally, with respect to security, the *options* field in the header of the IP datagrams supported two security options for labelling of sensitive information. The options are referred to as the Basic Security Option (BSO) and the Extended Security Option (ESO). Security labels consist of the classification level at which the datagram is to be protected (such as top secret and secret), the authorities whose protection rules apply to each datagram, and some extra security information (only for the ESO). There are also new labelling standards, the NIST Standard for Secure Labelling (SSL) and the DoD Standard for Common Security Label (CSL). A Commercial Security Option (CIPSO) has been proposed by the Trusted Systems Interoperability Group (TSIG) to meet commercial instead of military requirements.

However recently, there has been considerable work within IETF to develop security mechanisms for IP, as part of the IP Security Protocol Working Group (IPSEC). A security protocol in the network layer supporting authentication, integrity, confidentiality and access control is being developed. There are two specific headers that are used to provide security services in IPv4 and IPv6. These headers are IP Authentication (AH) (RFC 1826) and the IP Encapsulating Security Payload (ESP) (RFC 1827). The IP AH is designed to provide integrity and authentication without confidentiality to IP datagrams. The IP ESP is designed to provide integrity, authentication and confidentiality to IP datagrams. A key management protocol called the Internet Key Management Protocol (IKMP) is also being defined at the application layer.

The Secure Data Network System (SDNS) protocols have been developed within the framework of the OSI to support secure interaction between applications. They are also intended to provide secure communications to DoD and commercial data networks with a preference for the TCP/IP stack. The SDNS protocols essentially encapsulate the protocol data units in a "security envelope" with some protected header in front. The protected header may have security labels, sequence numbers along with addresses and headers of the specific protocol. An Integrity Check Value is appended to the PDU. SDNS defines two categories of protocols, namely the SP3 family [12] which resides in the network layer, and the SP4 family [10] which resides in the transport layer. There are four variants of SP3 and two variants of SP4.

The SP3 protocols provide connectionless network security services, primarily for the ISO CLNP protocol. Extensions to the basic protocol have been defined to allow its use for TCP/IP systems, and to enable its termination at intermediate points. This is achieved by encapsulating routing information in the protected header of the SP3 Protocol Data Units (PDUs). Depending on the format of the protected portion of its header, the SP3 protocol can operate in different addressing modes, namely SP3N, SP3A, SP3I and SP3D. SP3A is at the top of the network layer and it includes the source and destination NSAP addresses in the protected header. SP3I lies below the CLNP network sublayer and includes the CLNP header in the protected header. SP3D is similar to SP3I except that it lies below the DoD IP protocol. SP3N is identical to SP4E and is used only in the end systems.

The integration of SP4 within the transport protocol, allows access to all of the Transport Protocol control information. Thus the SP4 protocols permit the use of cryptographic techniques to provide data protection for transport connections or for connectionless-mode TPDU transmission. SP4C is closely integrated with the OSI connection-oriented services (ISO 8073), and SP4E provides support for connectionless-mode transport service (ISO 8602) and DoD TCP. That is, SP4C can be seen as a sublayer near the bottom of the transport layer. Hence a separate security association with a separate key is formed for each transport association, even when the transport connections are between the same transport entities. SP4E resides between the transport and the network layers. Hence it is dependent on the services of the transport layer for connection integrity.

In addition to these secure communication protocols, the SDNS project also defined a Key Management Protocol (KMP).

3.3.2 OSI Security

In terms of security, the work that directly addresses the security issues in the OSI Architecture is the Security Architecture document (ISO 7498-2) [9]. It is worth emphasizing that it only defines a skeleton for the provision of security, and does not provide any details as to how the security services be provided. It deals with security at an abstract level, defining a number of security services and mechanisms to support them, and does not describe specific security protocols. The Network Layer Security Protocol (NLSP) and Transport Layer Security Protocol (TLSP) are upgarded versions of the SDNS SP3 and SP4, standardised by the ISO for use with the OSI compliant network and transport layers. With respect to connectionless service, NLSP (ISO 11577) provides the same services as SP3, plus traffic flow confidentiality. In addition, NLSP addresses the protection of the connection-mode network service defined in CCITT X.213. This is not the case with SP3 which only deals with the connectionless aspect of the network service in terms of the ISO CLNP and DoD IP. Furthermore, NLSP supports in the connection mode in-band key distribution during connection establishment or within an on-going connection. TLSP (ISO 10736) is almost identical to SDNS SP4. In the application layer, there are several OSI standards that address security aspects such as messaging (X.400) and file transfer (FTAM). Some are at initial stages of development whereas others such as X.400 (1988 Recommendations) have specified a comprehensive set of security services and profiles.

4. Security for Broadband Networks

Recall that the broadband MAN/WAN technologies support both data networks as well as circuit-based networks such as voice and video traffic. These applications have real-time characteristics which affect the way they are set up and managed. The protocol stacks for broadband networks are somewhat different, and hence first it is necessary to look at suitable ways of incorporating security services within these protocol profiles. Furthermore, these networks have a separate out-of-band signalling path. Hence from security point of view, there is a facility to integrate the security management protocols such as the key management as part of the signalling phase in the Control Plane (C Plane) rather than in the User Plane (U Plane).

With these in mind, let us consider the issues relating to provision of security in such broadband networks. The first question that arises is where in the protocol stack should the security services be provided. There are several options.

Application-embodied Security

In this option, the functionality of each individual application has to be enriched in order to support security services. This approach may be useful when specialized application-oriented security is required. This will offer protection at the highest possible level in the stack.

Security at the Stack-level

In the OSI stack, end-to-end security could only be achieved above the network layer. This is because the information required for routing occurs at the network level and this information needs to be in plaintext form. Subsequently, security protocols designed for this type of networks such as the TLSP, the NLSP and the SDNS' SP3 and SP4, operate in and between the transport and the network layers. End-to-end security avoids the need to place any trust on the resources such as routers and intermediary devices which are not owned by the sender and receiver (organizations).

The applicability of these security protocols appears to be limited in the context of broadband networks. First, the routing in such broadband networks is done based on values provided within the data link layer instead of the network layer. For instance, with a Frame Relay network, the routing is based on the DLCI values provided within the data link layer. Therefore it is now possible to provide end-to-end security at a lower level. Second, as mentioned earlier, transport protocols such as the TCP and the OSI's TP0-4 can become bottlenecks in high speed environments. Given this, it is likely that they will be modified in

the near future by some light weight protocols. These will present an interface that will differ from the current ones for which the security protocols have been originally designed for.

Security at the Interface-level

There are three driving forces behind the provision of security at the interface level. Firstly, the layers comprising the access interface are always present independently of the supported traffic. Hence all the applications can use the security services offered by a security sublayer operating at this level. Secondly, both the user and the network operator must be given the choice to protect the traffic. The network operator has much fewer options by being restricted to provide security services within the boundaries of his domain. Thirdly, an internetworking device can act as a security service provider. It can effectively act as a "frontdoor-lock". Not all end systems may have or indeed need to have built-in security mechanisms. Furthermore, such secure internetworking devices can be used to translate and interpret different security policies between networks, e.g. between public and private networks.

Considering the network access interfaces shown in Figure 1, we have several options for the placement of security services within these interfaces. Let us now consider each of these options.

- First consider the placement of security at the physical layer. The physical layer strictly deals with the medium and the characteristics of the transmitted signals. Protection at this layer can only take place in the form of scrambling of signals, using an encryption device at each link. Such a solution is very limited and inflexible. To decouple security mechanisms such as encryption from the medium (e.g. coaxial cable, twisted pair) and the encoding scheme (e.g. 4B/5B, HDB3), it is necessary to perform encryption just before translation to the characteristics of the medium occurs, and immediately after line coding has been carried out. Otherwise, each network interface will require a distinct type of encryption device upon adoption of a different physical layer medium dependent sublayer. Such a technique is useful for protection against traffic flow confidentiality, for instance, in an exposed link between the customer premises and the network switch (e.g. the User-to-Network Interface (UNI) in the N-ISDN and B-ISDN).
- Another option is to integrate security into one of the network access interfaces. For instance, in the case of Frame Relay, security can be integrated within the DL-CORE sublayer. However such an approach often impacts the functionality of that interface. Even when a clear interface exists between two successive sublayers *within* the protocol profile of a broadband access interface, interoperability is likely to be severely downgraded with other technologies. Therefore, in practice, it is preferable to avoid such an approach.
- Another option is to place the security functions on top of the access interface. In this way, it is possible to support a wide range of security services at this level. For instance, in the case of Frame Relay, we can place security on top of the DL-CORE sublayer. In the case of a LAN, the IEEE 802.10 standard placed the security layer on top of the MAC layer. Such an approach is attractive for incorporating security in devices such as remote bridges and routers. In fact, in our view, this option of providing security at the top of the access interface represents the most effective way of providing secure LAN-to-LAN interconnections, which is one of the main drivers of public broadband services.

5. Security for Connection-Oriented Service

The rest of this paper is concerned with the demonstration of providing security services on top of the access interface by considering the connection-oriented Frame Relay networks. The connectionless SMDS service has been considered in [3].

In general, for data traffic, the connection-oriented service offers more efficient management of traffic than the connectionless service, by allocating resources within the network. Another advantage occurs when the data is to be transferred over long periods; in this case, the duration of the call set-up phase can be justified by subsequent savings in time.

Moreover, circuit-type traffic with low service requirements can also be users of the connection-oriented service. For example, poor quality voice and low scan video could make use of connection-oriented Frame Relay. However circuit-type traffic with stringent timing constraints could suffer severe degradation of service. It may be possible to use under certain circumstances, for instance, providing access to an ATM-based core network (with no congestion and the end systems supporting appropriate traffic shaping mechanisms). In general, they are better handled using fixed cells than using variable length frames.

5.1 Frame Relay

Frame Relay can be thought of as a lightweight descendant of X.25. Here, much of the sophisticated control functionality and facilities found in X.25 [1] are sacrificed for the sake of high speed data transmission. Moreover, identification of the virtual channel now takes place at the data link layer instead of the network layer as in the case of X.25. As a result, Frame Relay gives an order of magnitude improvement in network throughput over X.25.

There are two types of Frame Relay connections : *permanent virtual connections* (PVCs), and *switched virtual connections* (SVCs). The establishment, maintenance, and release of PVCs are subject to local management operations. On the other hand, signalling is required to manage SVCs. Dynamically allocated SVCs are more attractive than the PVCs which function as dedicated private lines. At present, the Frame Relay implementations are primarily PVC-based. This is due to both the complexity of the required signalling and its unavailability on the local loop [5]. There is a growing demand for products supporting SVCs. However, PVCs provide a good immediate solution for LAN to LAN interconnectivity applications.

The Frame Relay interface is based on the core functions of the LAP-F protocol. This protocol is defined in the CCITT Rec.Q.922, and it is also sometimes referred to as Rec.I441* (* stands for extended). LAP-F allows the existence of multiple instantaneous logical sessions (statistical multiplexing) within a single physical channel. The transferred service data units appear in the form of frames. An attached logical identifier (DLCI) with local significance is used to identify the virtual circuit this frame belongs to.

In general, the LAP-F is subdivided into two sublayers each implementing a different part of its functionality. The first sublayer is called DL-CORE, and it offers only Frame Relay services. It deals with the addressing of frames, detection of errors (but not with recovery of frames in error), and supports some basic congestion control. The second sublayer is called DL-CONTROL, and it implements the actual control functionality of LAP-F. It is strictly concerned with information included within the control field of the LAP-F frame. This sublayer offers reliable transfer of information enabling the acknowledgement of frames and the recovery of lost frames. The Frame Relay interface implements only the functionality of the DL-CORE sublayer.

In principle, the connection-oriented service offered by Frame Relay addresses either data or circuit type traffic. Simultaneous support of both types of traffic may also take place. For example, packetized voice and data can be transferred over the same virtual connection offered by the Frame Relay interface during a LAN to LAN interconnection.

In a Frame Relay network, DLCI values at the DL-CORE level are used to identify the communication path. Given this, all the routing information in a Frame Relay network is provided within the interface, that is, at a lower level than the network level. Consequently, the network layer may become redundant during the data transfer phase. The main tasks of the adaptation layer are to segment the resulting bitstream into small information units that the underlying technology can handle, and to preserve the required synchronization. In some technologies such as DQDB and B-ISDN, the adaptation layer forms part of their interfaces. This is not the case with Frame Relay, where it has to be provided on top of its interface.

5.2 Secure Frame Relay Connections (SFRC) Layer

The placement of security within the Frame Relay interface can logically occur at the physical layer, or can be integrated into the DL-Core sublayer, or can be at the top of the DL-CORE layer. Following the discussions in Section 4, it is proposed that the Secure Frame Relay Connections (SFRC) layer operates on top of the DL-CORE sublayer (See Figure 2).

Note that SFRC is different to the IEEE 802.10 SDE layer in that it should be able to cope with situations where there is no MAC sublayer. Consider for instance an user equipment accessing the Frame Relay interface either directly or by being connected to it via an ISDN interface. Here the MAC sublayer is absent and a data link protocol (e.g. LAPF) is used to pass the traffic over a multidrop line shared by several terminals. Another example may be a video-conferencing application between two studios over an ATM-based core network, where an internetworking device implementing a Frame Relay interface is providing access to the ATM network. An adaptation layer can be used to handle the bit streams from the video conference application and to ensure synchronization. Once again a MAC layer is absent in such a situation (See Figure 3).

Hence the need for SFRC layer to protect the different types of Frame Relay traffic. The SFRC should be able to support the security services required during both the call control phase and the data transfer phase of the Frame Relay.

The SFRC sublayer comprises one or more entities, each providing security services to an individual frame relay virtual connection. Communication between SFRC entities located in remote systems is achieved in terms of the SFRC protocol. The message units related to a connection are exchanged between the SFRC and its adjacent (sub) layers via points identified by the endpoint identifiers (CEIs).

The services offered by the SFRC are specified by describing the information flow to the layer immediately above (SFRC-user) and to the layer below (DL-CORE) in terms of service primitives. By having the SFRC sublayer operate on top of the Frame Relay interface in a transparent way, the primitives used across the service interface of the SFRC sublayer and the higher sublayer are identical to those supported by the DL-CORE sublayer. Parameters associated with the SFRC-DATA primitives are identical to those found in the corresponding DL-CORE primitives. The SFRC sublayer only processes the DL-CORE-User data field of a primitive; all other parameters are transferred transparently. These parameters are defined in Annex C of CCITT Rec.I.233.

5.3 SFRC Layer Security Services

The SFRC layer supports the following security services : data origin authentication, access control, connection confidentiality and connection integrity without recovery. In addition to these, associated with the call setup phase of the SVCs, support is provided for peer authentication, establishment of a secret dialogue key, and release of a connection in an authorized manner. Furthermore, the protocols between the SFRC layer managers support secure negotiation of a session between their local SFRC entities, dynamic activation and deactivation of the negotiated security mechanisms during the data transfer phase, and renegotiation of the secure connection while the connection is still in place.

The structure of a SFRC protocol data unit is shown in Figure 4. It consists of four parts: a clear header, a protected header, user information field, and a trailer. Each of these PDU parts is further subdivided into a number of fields.

The Secure Connection Identifier (SC-ID) value associates the SFRC PDU with a secure frame relay connection at the destination frame relay interface. We recommend the use of the DLCI values as SC-IDs. Connection confidentiality is provided by encrypting the User Info. Connection integrity without recovery is provided by the inclusion of an Integrity Checksum ICV. The Protected Header contains the Security Label information as well as information regarding padding. The contents of this header can be protected for both confidentiality and integrity. Data origin authentication is provided by guaranteeing the association of frames with the virtual path over which the frames are transferred. This is done by having a unique shared key between the SFRC entities at the end points of a Frame Relay connection. A

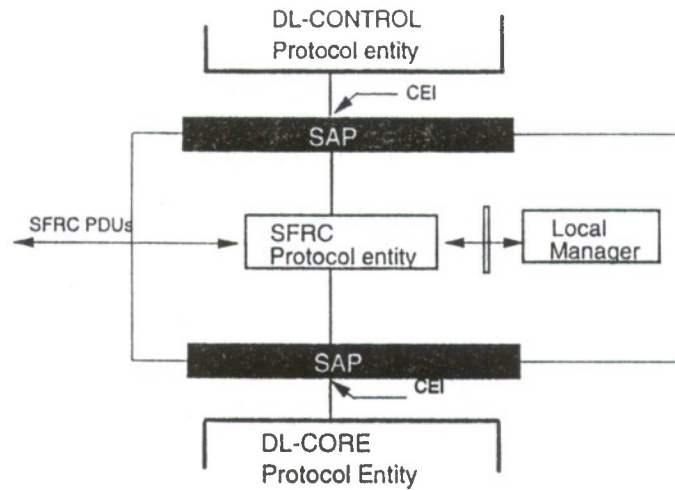


Fig. 2: Secure Frame Relay Connections (SFRC) Layer

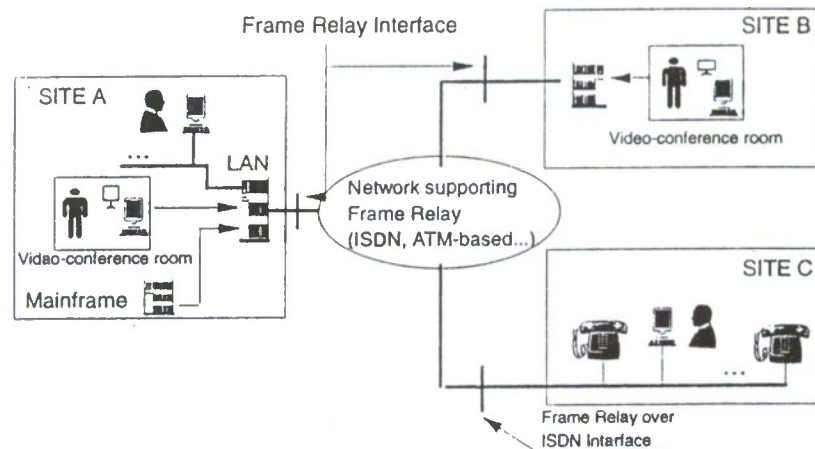


Fig. 3: Topology restricting the use of SDE

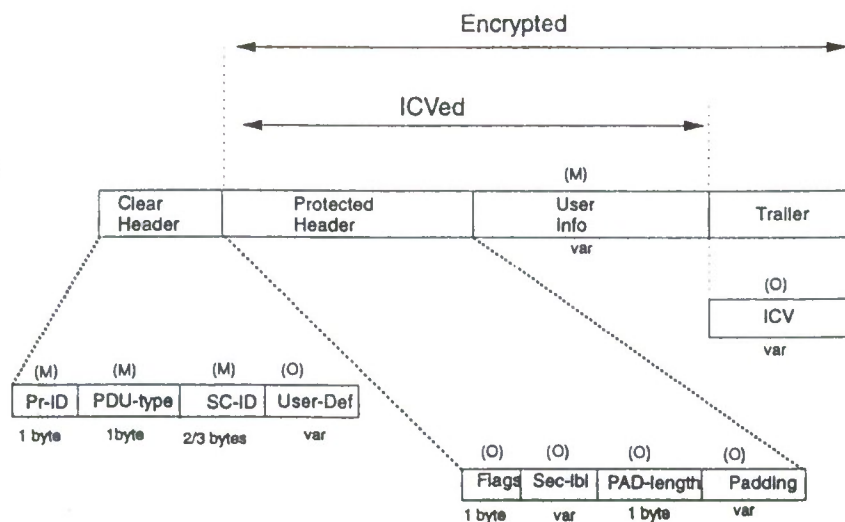


Fig. 4: SFRC Protocol Data Unit

fuller description of the security services can be found in [14]. The access control mechanism determines which SFRC entities can communicate with each other, which in turn determines which entities can establish a secure conversation key. We discuss the secure call setup in Section 5.

Just a brief explanation for not including the connection integrity with recovery and traffic flow confidentiality security services.

Connection integrity with recovery service : In contrast to SP4 [11] and TLSP [12] which can access the sequence numbers of the transport layer, the SFRC has no access to such information in the DL-CORE sublayer. It is for this reason the ISO 7498-2 considers the provision of the connection integrity with recovery service at the transport layer and not at the network layer. Hence deletion of a protected SFRC PDU will only be detected by a higher sublayer (e.g. DL-CONTROL or transport layer), and retransmission will be requested. But in this case, the SFRC sublayer has no knowledge of this event. However, inclusion of an invalid PDU will be detected by the SFRC even when the attacker has knowledge of the current sequence number. This is because the attacker does not have access to the integrity and/or encryption key(s) used by the SFRC.

Traffic flow confidentiality service : Two security mechanisms can be used to support this service : routing control and traffic padding. The first case allows the routing to be determined by the sensitivity of the transferred information. This is only feasible when the intermediate nodes are able to exercise access control. In Frame Relay, it is either the signalling (SVCs) or the network operator (PVCs) which determines the routing by establishing the appropriate virtual circuits. In principle, SFRC can be used by the intermediate nodes to support access control. However, as SFRC has been designed to operate as an end-to-end security protocol it has been decided not to support this functionality. Alternatively, traffic padding can be used to provide traffic flow confidentiality. This is done by transmitting encrypted dummy frames while there is no demand for real traffic transfer. The enforcement of this security mechanism by the SFRC sublayer suffers from the following drawbacks : (a) Continuous transmission of information causes loss of bandwidth, which may have an adverse financial impact. (b) The inclusion of such a service can make the functionality of the sublayer more complex. A timer has to be supported, and synchronization with the adjacent layers needs to be ensured. Otherwise the generation of dummy frames will prevent SFRC from processing outstanding primitives and degradation of service will be inevitable. Perhaps more importantly, the observation of traffic patterns may not be a serious threat in many situations.

5.3.1 SFRC Management Information Base (MIB)

The processing of frames by the SFRC sublayer is controlled by security management information. This information is stored in the form of managed objects as part of the Management Information Base (MIB), called the SFRC-MIB. Each managed object is defined by its attributes, a set of management operations, and internal actions and notifications which the managed object translates into management messages.

Objects stored in the SFRC-MIB are identified as instances of either *system* objects or *secure connection* objects. The system objects apply to the entire system regardless of any individual connection. But an instance of a secure connection object, known as SC-entry is tied to a specific Frame Relay connection protected by a local SFRC entity. Therefore, the same SC-entry must be supported in the local SFRC-MIBs of the two SFRC entities involved in a protected Frame Relay connection.

The System objects are divided into the following six categories : Confidentiality, Integrity, System-Security-Labels, Security-Profile, Domain, and Trusted-Authorities. Secure connection object deals with the following information of the connections : security profile in use, user information length, local SC-ID, peer SC-ID, encryption and decryption keys, signing keys, security labels, and identifiers of confidentiality, integrity and signing algorithms.

5.4 Secure Call Establishment

Let us now look at the call establishment phase and consider the services required to establish secure switched connections (SVCs).

Figure 5 illustrates the flow of the signalling messages between the originating and the terminating users. User information is exchanged during set-up and clearing phases of the call, as part of the User-User Information (UUI) element. The UUI field has been used to transfer security related information required in the establishment of secure SVCs.

During the call establishment phase, all three messages carrying the UUI (SETUP, ALERTING and CONNECT) have global significance. They can be used to pass the information required for mutual peer authentication and establishment of secret keys on an end-to-end basis. We only use the SETUP and CONNECT messages in our system, as the ALERTING message is in principle not mandatory. Before describing the protocol, it is necessary to look at the structure of the UUI element.

The structure of the call establishment message envelope is shown in Figure 6. The call establishment Services field indicates the security services supported by the UUI element. These services include mutual or one-way authentication and key distribution. The value of this field is used to determine which party has to authenticate and/or participate in the creation and distribution of the secret conversation key. This is followed by related information for these security services. For instance, this can be used to specify alternative mechanisms for the provision of a security service. Then comes the actual call establishment message, followed by an Integrity Checksum Value which is a function of the contents of the envelope and the message.

5.4.1 Secure Call Setup Protocol

The secure call setup protocol establishes secure SVC with mutual authentication of the parties and with the agreement of a secret key between them that is available for use in subsequent data transfer.

Note that users, application processes, protocol entities, and devices are amongst the parties that may authenticate one another. For instance, the call setup may be activated to authenticate the person involved in a telephone conversation, the client application process setting up a connection to access a remote server, and even the device it runs on. Also the timeliness (freshness) of the authentication procedure needs to be guaranteed. This aspect leads to problems when applying certain protocols. In particular, a natural way of providing freshness would be to use challenge-response protocols. Unfortunately, such mechanisms require at least three exchanges when it comes to mutual authentication. In this context, the exchange of the SETUP and CONNECT messages alone cannot accommodate such a scheme, except when only one-way authentication is required. Mechanisms requiring timestamps can be used, as they require only a two-way handshake; however it becomes necessary to maintain some form of synchronized clocks, which is not trivial.

There are a number of existing protocols that can be used to provide mutual peer authentication and key distribution between the calling and the called parties. Below we present one such protocol based on X.509 [15]. We assume the existence of Certification Authorities (CAs); we do not address the interactions between the communicating parties and the CAs (and the interactions between the CAs). Consider the establishment of a SVC between two devices A and B. We use A_S to denote A's private key and A_P to denote A's public key.

- When A wishes to be authenticated by B, it sends its certificate and a new authentication token ($A_S[\text{hash}(t_A, r_A, B, \text{Data}, \text{Key}_A)], B_P[\text{Key}_A], t_A, r_A, \text{CallReference}$). These form part of the UUI element in the SETUP message in the call establishment envelope.

The token contains a timestamp t_A and a random number r_A to prove its freshness and to protect against replay attacks. The *Data* portion contains the Calling Party Number, Calling Party Sub-address and the Call Reference information elements. The *Data* portion is signed using the private key of A (A_S). The *Data* portion also appears separately as plaintext within the SETUP message. The only exception is the *CallReference* element. It only identifies the call at the local user to network interface to which the particular message applies. Hence *CallReference*, t_A and r_A are sent to enable B to calculate the token's hashed value. The *Service* field in the envelope identifies key distribution. The Key_A is encrypted under B's public key to protect its confidentiality.

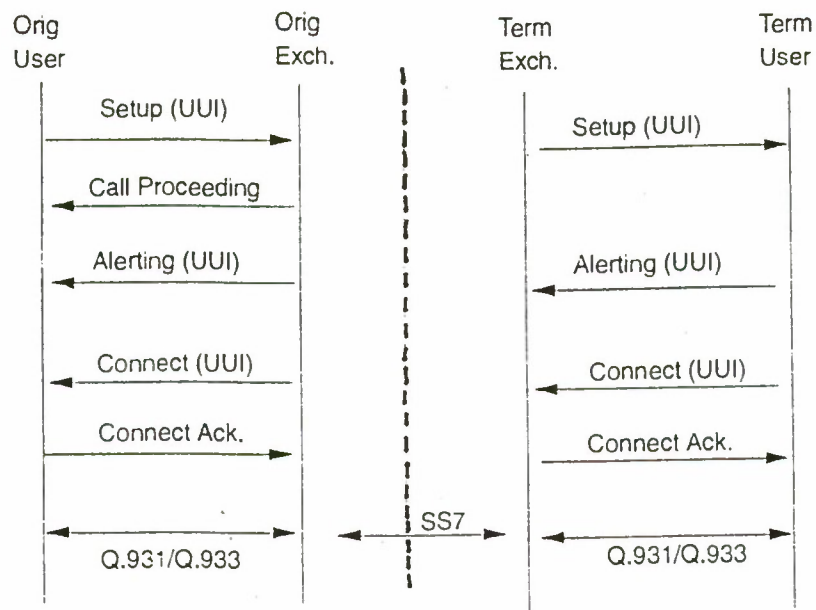


Fig. 5: Call Setup Message Flows

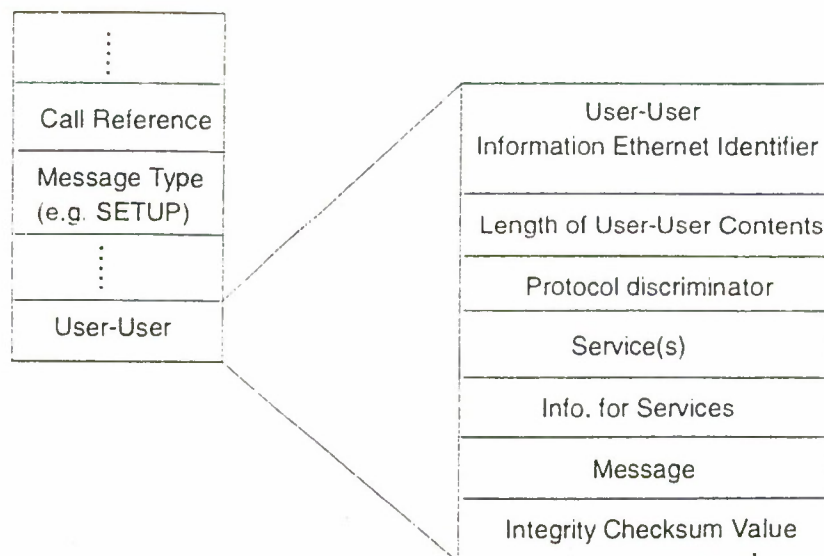


Fig. 6: Call Establishment Envelope

- Upon reception, B verifies the certificate (using its copy of the public key of the Certification Authority). It then uses the public key of A recovered from the certificate to verify the authentication token. This will allow B to determine whether this SETUP message comes from A and if it is intended for B. B also checks the timestamp t_A to establish whether this message is within the allowed time window (for freshness). Finally, B retrieves Key_A using its private key.

B then sends to A its own certificate, and the authentication token

$(B_S[hash(t_B, r_B, A, r_A, Data, Key_B), A_P[Key_B], t_B, r_B, CallReference])$. These are included within the UI element of the CONNECT message. The authentication token contains a timestamp t_B and a random number r_B to protect against replay attacks. A's random number r_A is included in the reply.

- When A receives the CONNECT message, it verifies B's certificate using the public key of Certification Authority. Then A checks that the timestamp t_B and the random number r_B for freshness, and verifies whether the authentication token in the reply was actually generated by B, and is intended for A. It also retrieves Key_B .
- There are three possible options with respect to the generation of the secure conversation key. Either A generates it and forwards it to B, or B generates it and sends it back to A, or the conversation key could be a combination of Key_A (generated by A) and Key_B (generated by B). A and B can now form the secret conversation key, using Key_A and Key_B , which can be used to secure subsequent communications.

Acknowledgements : The authors would like to thank Hewlett-Packard Labs.UK. The authors would also like to thank the anonymous referees for their valuable comments.

References

- [1] CCITT Recommendations X.25, Fascicle VIII.2, IXth Plenary Assembly 1988 (Blue Book).
- [2] Bellcore : "Generic System Requirements in support of Switched Multi-megabit Data Service", Bellcore Technical Reference TR-TSV-000772, May 1991. Also ESIG TS-001/93, SMDS Service Definition and Subscriber Network Interface, March 1992.
- [3] V.Varadharajan, "Securing Local Area and Metropolitan Area Networks : A Practical Approach", Proc. of the 18th National Information Systems Security Conference, Baltimore, Oct.1995.
- [4] IEEE 802.6, Distributed Queue Dual Bus (DQDB) Sub-network of a Metropolitan Area Network (MAN), Dec.1990.
- [5] *Digital Subscriber Signalling No.1 Signalling Specification for frame mode bearer service*, CCITT Recommendation Q.933.
- [6] Elke Gronert, *MANs make their mark in Germany*, Data Communications, Vol.38, No.11, Nov.1990.
- [7] Andre Danthine, *Espit Project OSI 95 : New Transport Services for High Speed Networking*, Computer Networks and ISDN Systems, Vol.25, Nov.1992.
- [8] IEEE, Standard for Interoperable Local Area Network (LAN) Security (SILS) - Part B - Secure Data Exchange (SDE), IEEE 802.10b, 1992.
- [9] ISO, Information Processing Systems - Open Systems Interconnection Reference Model - Security Architecture, ISO 7498/2, 1988.

- Upon reception, B verifies the certificate (using its copy of the public key of the Certification Authority). It then uses the public key of A recovered from the certificate to verify the authentication token. This will allow B to determine whether this SETUP message comes from A and if it is intended for B. B also checks the timestamp t_A to establish whether this message is within the allowed time window (for freshness). Finally, B retrieves Key_A using its private key.

B then sends to A its own certificate, and the authentication token

($B_S[hash(t_B, r_B, A, r_A, Data, Key_B), A_P[Key_B], t_B, r_B, CallReference]$). These are included within the UUI element of the CONNECT message. The authentication token contains a timestamp t_B and a random number r_B to protect against replay attacks. A's random number r_A is included in the reply.

- When A receives the CONNECT message, it verifies B's certificate using the public key of Certification Authority. Then A checks that the timestamp t_B and the random number r_B for freshness, and verifies whether the authentication token in the reply was actually generated by B, and is intended for A. It also retrieves Key_B .
- There are three possible options with respect to the generation of the secure conversation key. Either A generates it and forwards it to B, or B generates it and sends it back to A, or the conversation key could be a combination of Key_A (generated by A) and Key_B (generated by B). A and B can now form the secret conversation key, using Key_A and Key_B , which can be used to secure subsequent communications.

Acknowledgements : The authors would like to thank Hewlett-Packard Labs.UK. The authors would also like to thank the anonymous referees for their valuable comments.

References

- [1] CCITT Recommendations X.25, Fascicle VIII.2, IXth Plenary Assembly 1988 (Blue Book).
- [2] Bellcore : "Generic System Requirements in support of Switched Multi-megabit Data Service", Bellcore Technical Reference TR-TSV-000772, May 1991. Also ESIG TS-001/93, SMDS Service Definition and Subscriber Network Interface, March 1992.
- [3] V.Varadharajan, "Securing Local Area and Metropolitan Area Networks : A Practical Approach", Proc. of the 18th National Information Systems Security Conference, Baltimore, Oct.1995.
- [4] IEEE 802.6, Distributed Queue Dual Bus (DQDB) Sub-network of a Metropolitan Area Network (MAN), Dec.1990.
- [5] *Digital Subscriber Signalling No.1 Signalling Specification for frame mode bearer service*, CCITT Recommendation Q.933.
- [6] Elke Gronert, *MANs make their mark in Germany*, Data Communications, Vol.38, No.11, Nov.1990.
- [7] Andre Danthine, *Esprit Project OSI 95 : New Transport Services for High Speed Networking*, Computer Networks and ISDN Systems, Vol.25, Nov.1992.
- [8] IEEE, Standard for Interoperable Local Area Network (LAN) Security (SILS) - Part B - Secure Data Exchange (SDE), IEEE 802.10b, 1992.
- [9] ISO, Information Processing Systems - Open Systems Interconnection Reference Model - Security Architecture, ISO 7498/2, 1988.

- [10] SDNS Program Office, *Security Protocol 4 (SP4)*, SDN.401, Revision 1.2, July 1988.
- [11] ISO/IEC DIS 10736, Transport Layer Security Protocol, TLSP, Dec.1991.
- [12] SDNS Program Office, *Security Protocol 3 (SP3)*, SDN.301, Revision 1.3, July 1988.
- [13] ISO/IEC CD11577, Network Layer Security Protocol, NLSP, Nov.1991.
- [14] P.Katsavos, V.Varadharajan, *A Secure Frame Relay Service*, Computer Networks and ISDN Systems, Vol.26, pp1539-1558, 1994.
- [15] CCITT Recom X.509, The Directory Authentication Framework, Geneva 1989 (revised 1992).

A CASE STUDY OF EVALUATING SECURITY IN AN OPEN SYSTEMS ENVIRONMENT

Authors:

Daniel L. Tobat , TASC, formerly The Analytical Sciences Corporation,
12100 Sunset Hills Road, Reston VA 22090 Tel: 703.834.5000
Fax: 703.318.7900 E-mail: dltobat@tasc.com Web: www.tasc.com

Errol S. Weiss, Technical Director, Science Applications International Corporation's
Center for Information Protection, 8301 Greensboro Drive, McLean, VA 22102
Tel: 703.556.7366 Fax: 703.448.7360 E-mail: errol@cip.saic.com

Abstract

The goal of this paper is to describe a case study of a computer security evaluation effort conducted on a system known as the Office Automation Network (OAN). The OAN is representative of many of today's networked systems by being a heterogeneous mix of system components connected to open systems such as the Internet. The OAN differs from typical systems in that security was a design and implementation objective, and that it was subjected to an extensive six month evaluation effort by an experienced vulnerability testing team. The vulnerability testing yielded some surprising results which demonstrated that it is possible in today's environment to have an Automated Information System (AIS) connected to open systems such as the Internet and still have an effective security posture.

Introduction

The large scale networking of Automated Information Systems (AIS) on a worldwide basis has implications for the information systems security field which are only now becoming widely recognized. The ability of current information technology to network and interconnect systems has in most cases far outpaced the ability to protect these networks. In the interest of interoperability, widespread sharing of information and doing more work with a smaller, more technically agile workforce, the push is on to increasingly network systems and to connect these AIS to globally interconnected "network of networks" such as the Internet. The increased efficiency of AIS networking has generally come at the price of increased vulnerability of systems and information to attack. An AIS connected to open systems environments such as the Internet can be accessed worldwide thus exposing systems to a wide range of potential security threats. In the networked environment of operational systems, not only is it more of a challenge to protect systems, it also becomes increasingly difficult to determine the security posture of a networked system. As technology increasingly pushes open systems environments, and as computer and communication technologies continue to converge, it becomes difficult for even system administrators and technical personnel to know the full extent of individual system boundaries and capabilities. This technological convergence often introduces new vulnerabilities into the overall information infrastructure which present potential intruders with more opportunities to target a wider range of information.

System Description

The OAN has its genesis in the early 90's as an effort to consolidate the architecture and technology of various segmented LAN's which had been separately implemented by different offices. From its inception, the OAN design recognized a need to provide two distinct and separate network segments linked by a central backbone: a "low" side offering users open access to the Internet and a "high" side for users with requirements to protect more sensitive information, yet still requiring shared information services with "low" side users. The OAN is a large scale, general purpose office automation environment, based on a Microsoft Windows-NT client-server architecture encompassing over 1,500 workstations, 30 servers and an external gateway on a Fiber Optic Digital Device Interface (FDDI) backbone ring, with thin-net Ethernet distribution to most client workstations. General office automation services such as E-mail, word-processing, spread sheet and database programs are provided to a user population of approximately 2,000. As shown in figure 1, the OAN is divided into a "high side" segment and a "low side" segment which are essentially mirror images of each other. While the fiber optic cable, servers and gateways are physically separated they are connected through a bi-directional E-mail guard. The E-mail guard is hosted on an Intel-based platform running the SCO/CMW trusted operating system. The OAN's Internet connection is hosted by a Digital Equipment Corporation (DEC) VAX 6310 running the VAX/VMS operating system. A separate DEC VAX 6310 hosts the E-mail guard on the high side of the OAN. In addition to the E-mail guard, the Network Monitoring Stations are the only network elements running the UNIX operating system, which consist of Sun SPARC workstations running the Cabletron Spectrum Network Management Tool Suite. A security policy is in place which limits off network access and permits limited E-mail capability to traveling users through the use of temporary accounts on

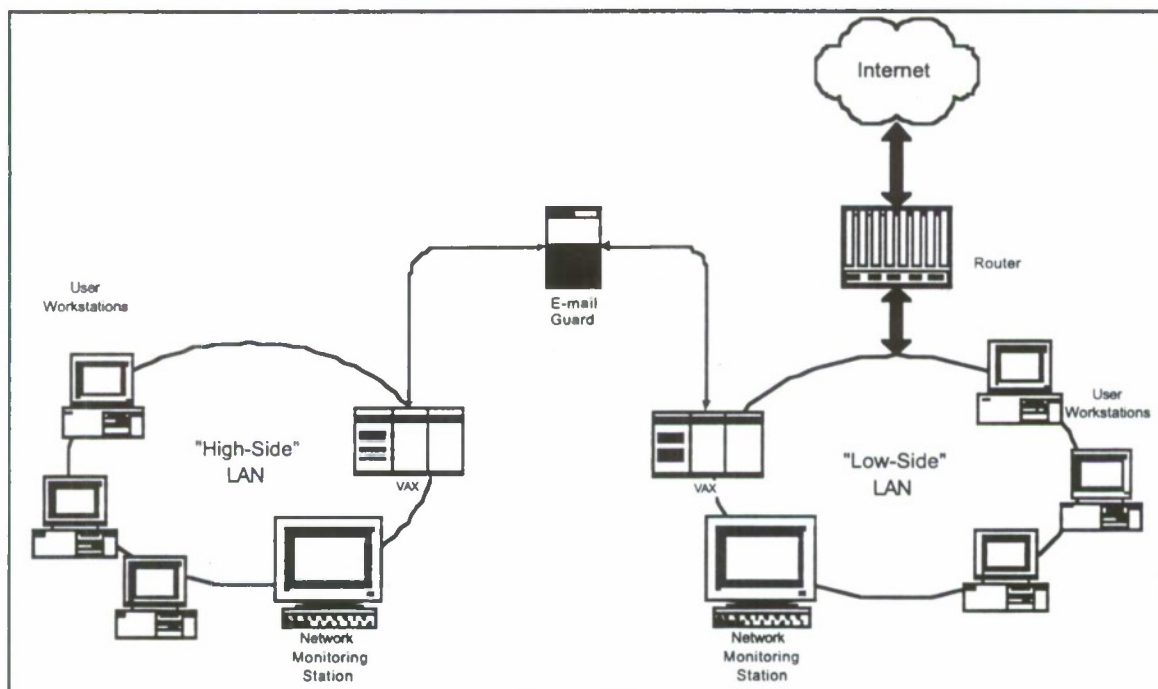


Figure 1. OAN Topology

the DEC/VAX system using static log-in/password procedures. The OAN was undergoing continual evolution during the six month course of this evaluation, such as upgrading the VAX 6310 hosts to DEC Alpha hosts which run both VAX/VMS and Windows. Packet filtering is employed by the OAN's TIMEPLEX Time/LAN 100 routers as a "firewall" technique. While not implementing a commercial offering of a "firewall," the combination of system features does in fact meet the definition of a "packet filtering firewall" according to the definitions developed by the National Institute of Standards and Technology (NIST) [1].

Key Component Descriptions:

This section provides additional detail on some key OAN components.

Windows-NT: This is a modular network operating system, which has been rated by the Trusted Products Evaluation Program (TPEP) at a "C2" level of security. The "C2" rating includes the ability of auditing to allow security related events to be recorded and monitored, the implementation of Discretionary Access Controls (DAC) and requiring Identification and Authentication (I&A) through the use of a mandatory log-in process to access the system. OAN servers run Windows-NT Server V3.51, while OAN clients use a mixture of Windows 3.1, Windows for Workgroups 3.1.1 and Windows-95 operating systems.

E-mail Guard: This component consists of a bi-directional E-mail guard that passes electronic messages and attachments between the low and high side LAN segments, and consists of a 486DX-50 MHz Intel platform running Santa Cruz Operating Systems (SCO) UNIX as the underlying operating system. The guard uses the "B1" TPEP rated Compartmented Mode Workstation (CMW) software package along with a custom designed user interface. The "B1" rating indicates the ability to support more restrictive security features than the "C2" rating such as the use of Mandatory Access Controls (MAC). The custom software package implements OAN security policies such as while high side users can send E-mail to low side users, they are not allowed to send E-mail to the Internet host. Low side users can send E-mail to high side users as well as out through the Internet. E-mail messages are encapsulated and signed to provide integrity. Prior to being deployed on the operational network, the E-mail guard underwent a security testing profile in a laboratory environment as a risk reduction technique.

The Challenge

While the OA-LAN had been designed and implemented with security as an objective, it had not been the subject of vulnerability testing by outside security experts. The widespread recent publicity associated with the "information warfare" concept, led to the commissioning of a non-trivial vulnerability testing effort to determine the security posture of the OA-LAN. Accordingly, a team of five personnel, with expertise across the range of OA-LAN systems and a combined vulnerability testing experience level of 33 years was put together to evaluate the OA-LAN. Because the OA-LAN was an operational system supporting thousands of users on a daily basis, it was not practical to exhaustively test for every potential vulnerability or to conduct denial of service attacks. While exhaustive vulnerability testing in a lab type environment is possible with development systems, large scale operational systems are usually too expensive to duplicate in a laboratory, and the consequences of many denial of service attacks can be difficult to predict. The principal objective behind this evaluation was to replicate the threat environment faced by the target system and which is summarized below. Any system connected to the Internet is susceptible to both external and internal threats and as such, should be subjected to periodic evaluations to determine its security posture. Performing a computer security evaluation on the OA-LAN was a challenge for several reasons. Traditionally, the majority of systems evaluated by the computer security community are UNIX systems. By contrast, the OA-LAN was a mixture of operating systems including Windows-NT, VAX/VMS and UNIX. The need to evaluate a broad range

of systems led to an extensive survey phase to fully train the team in the various technologies necessary and to set up a small scale mockup in a lab to explore and determine potential avenues of exploitation.

The Threat

For any system connected to the Internet, there is a very real threat from hackers, which will be defined here as a computer based intruder with no legitimate access on a system, and is also the term by which most computer intruders call themselves [2]. The Internet most closely resembles a global information superhighway, which connects over 35 million users through over 9 million hosts, linked by over 240,000 networks in 135 countries worldwide [3]. The Internet and its underlying Transport Control Protocol/Internet Protocol (TCP/IP) architecture were not designed to be secure. The phenomenal success of the Internet, in combination with the presence of unethical users has aggravated deficiencies to the extent that any system connected to the Internet risks inevitable break-in attempts. Several organizations, such as the Defense Information Systems Agency (DISA) Automated Information Systems Security Support Team (ASSIST) which track Internet intrusions on Department of Defense (DoD) systems, indicate at least one new intrusion attempt per day is now reported. In addition, the ASSIST regularly tests DoD sites with Internet connectivity for well known vulnerabilities which appear in Computer Emergency Response Team (CERT) bulletins. As of October 1994, over 88% of 8900 tested systems were easily penetrated, and 96% of the system penetrations went undetected [4]. The overall conclusion is that any system connected to the Internet can expect to be the target of repeated intrusion attempts and that many of these systems lack rudimentary levels of security. The majority of the widely publicized "information warfare" risk has focused on the external threat posed by hackers. Today's hackers include experienced, technically sophisticated intruders who have even published price lists for their services, and are willing to perform criminal activities for financial gain [5]. Hackers rely on a loosely organized, yet highly competitive computer "underground" for information exchange. Hackers usually begin by gaining admission to "entry level" groups of lower skilled members, and work their way into smaller groups of more sophisticated "elite intruders" by hacking systems and providing results as a proof of their expertise. In addition to the external hacker, all AIS have an internal threat due to the possibility of unscrupulous users, that is an individual with some degree of legitimate access to the system who performs unauthorized actions. Also worthy of note is the "disgruntled postal worker" syndrome, that is an employee who is losing their job as a result of workforce reductions and attempts to "get even" by sabotaging systems on which they have access. Data reported by the National Center for Computer Crime show that upwards of 85% of reported successful intrusion attacks on public networks are conducted or actively assisted by insiders [6]. In order to perform thorough vulnerability testing on an operational system, one must evaluate both the susceptibility of the system to both the external hacker as well as the "close in" unscrupulous insider.

Testing Approach

The key objective of the testing was to replicate the threat environment by performing various internal as well as external tests, in order to determine the security posture of the OAN. To simulate the external threat, the test team would use a series of standardized tools and scripts to mount attacks remotely across the Internet against the target system. A "standard hacker" tool suite would be run remotely against the OAN, and in addition an "intruder test team" was established to perform on-site internal testing. Internal threat testing usually focuses on the possibility of a user exploiting network functions in an attempt to "break out of the box" and perform unauthorized functions which are normally restricted to system administrators. For this evaluation, the intent was for the intruder test team to

replicate a "best shot" effort by an "elite intruder" hacker group operating with the premise of some on-site access in a two phased effort. In the initial phase, the test team would set up covertly on site, and attempt various "close in" technical exploitation methods without the knowledge of system administrators. During the subsequent phase, "social engineering" tactics would be used in an attempt to obtain user account passwords which would then be used in an attempt to subvert the system security policy from within, and finally various audit activities would take place with the knowledge and assistance of system administrators. The clearly expected result of devoting an experienced computer security testing team in an intensive six month long effort, is to achieve a successful system "break in" and demonstrate how the system could be exploited by outside hackers and or unscrupulous insiders. The experience of the team in past vulnerability testing efforts above led to expectation of a successful penetration with the only major question being the degree of difficulty to achieve intrusion, and whether that intrusion would be detected. Due to the many different computer systems present in the OAN, the five person test team underwent an extensive survey phase, obtaining all possible document on security features and vulnerabilities of the target system, received training on the relatively new Windows-NT operating system, and set up limited mock ups in a laboratory setting to explore different avenues of attack.

Vulnerability Testing Results

After an intense five month long preparation period, both internal and external testing was conducted over a one month period. Key testing activities are graphically depicted on figure 2. The results of these testing activities directed against the OAN were surprising. In an environment, where finding computer security vulnerabilities sometimes of an extensive nature is all too typical, the concluding results of this evaluation was that the OAN had an effective security posture. The "standard hacker" tool suites discovered no vulnerabilities while remotely testing the system. During the initial on-site testing phase, the on-site intruder test team was able to surreptitiously tap into the low side of the OAN by splicing in an extra 50 feet of cable to a "thin net" segment. The intruder test team then used five vulnerability testing stations on the added 50 feet of cable to conduct extensive "insider" testing. However, only minor configuration issues were discovered and the system could not be exploited further. Many of the team's hacking attempts were detected and reported by an OAN auditing team. During the second phase of the on-site testing, software auditing tools and wardialing were used to test for the presence of unauthorized modems, fax boards, and off-network access. A random sample of OAN workstations found a high rate of compliance with the system security policy. The intruder test team also attempted insider attacks against the E-mail guard in an attempt to gain unauthorized access to the high side of the OAN. The team found that the E-mail guard provides an effective barrier against attempts to access the high side. The test team reported that the auditing, monitoring and reporting procedures used on the OAN were excellent. The consensus of the test team was that the OAN had succeeded in achieving an effective security environment that was highly resilient to intrusion attempts. In view of the relatively poor security on many of the systems connected to the Internet, the OAN thus represents a model of how to implement an effective security posture in an open systems environment. This surprising result led to an intensive analysis to determine why the OAN had such an effective posture.

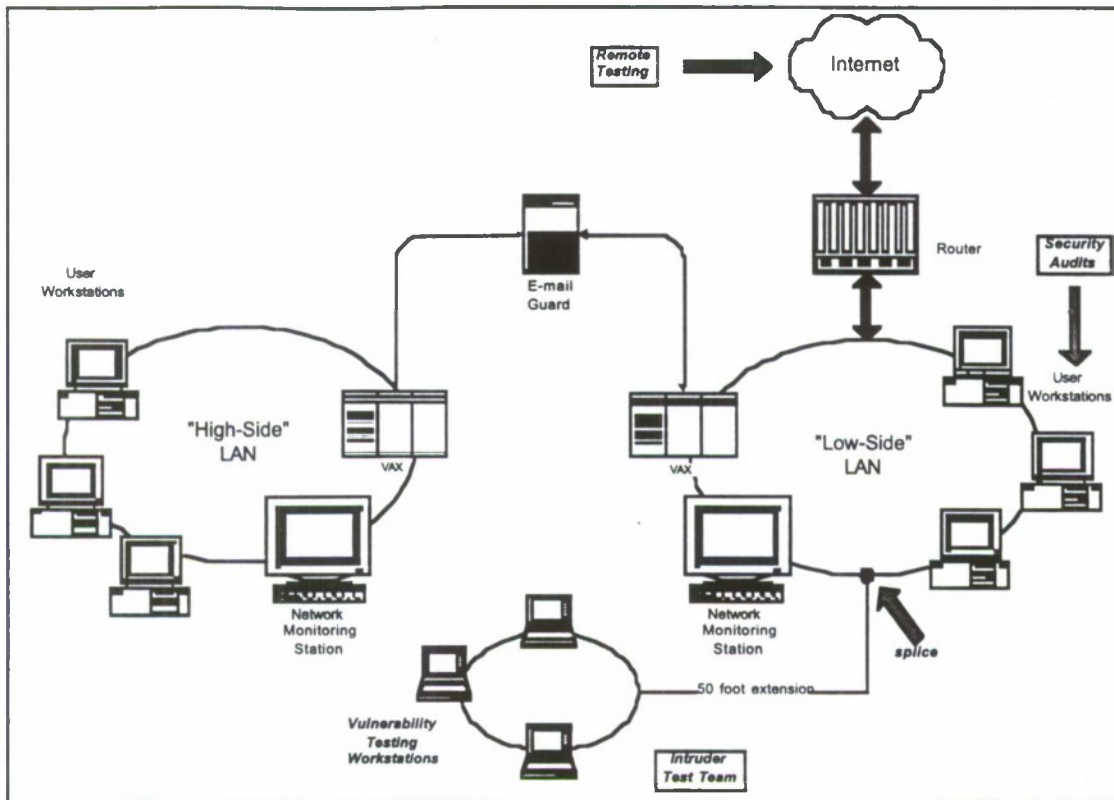


Figure 2. OAN Vulnerability Testing Activities

The unexpected result of finding a system so resilient to intrusion attempts led to an analytical effort to identify the reasons for the OAN's successful posture which led to this atypical outcome. The consensus of this analysis was that the OAN represented a system in which all the key items were present to ensure a strongly postured system including effective security policies, contributing technical factors and the allocation of sufficient resources to effectively secure this system. These specific factors are outlined in detail below.

Effective Security Policies:

The security policies employed on the OAN reduced risks by strictly limiting the outside connections allowed on the system. Off-network access is carefully controlled. No modem or fax capabilities are permitted and TCP/IP stacks installed on workstations are carefully implemented and frequently monitored to reduce risks. The E-mail accounts for roving users are temporary accounts which only have limited capabilities and are hosted on the DEC VAX processor. These policies are not only carefully designed to limit risks but are also effectively enforced. The on-site audit of a random number of workstations by the intruder test team confirmed that these policies are indeed followed by system users. In addition, the OAN policy of putting system administrators on notice that vulnerability testing could occur at any time without their knowledge is a highly commendable procedure. Note that while the OAN vulnerability testing was a "no notice" event to system administrators and the audit team, it was fully coordinated with site security personnel.

Contributing Technical Factors:

Several factors of a technical nature contributed to the OAN's security posture. One major factor was that the OAN was designed with security as an initial objective and was not an "add on" after the system was put together. Experience demonstrates that designing security in from the start is the most effective method to secure a system. The well thought out nature of the OAN's topology contributed to the system's resiliency to intruders. The other contributing factor was the use of the Windows-NT operating system. One of the most effective security features of Windows-NT was the selective use of encryption to protect packets containing log-in and password information, thus effectively limiting the exploitation of the system, even by intruders directly connected to the OAN. The tools used by both hackers and the vulnerability testing community are predominantly based on the UNIX operating system. Because of the widespread use of UNIX, it has been commonly used in multi-user operational environments and has long been the traditional target of computer based intrusions. Since the OAN does not rely on UNIX for the most part, most computer vulnerability tools and methods are ineffective at present. Note this is a transient advantage that will erode over time as the number and lucrativeness of attacking Windows-NT based systems increases, invariably tools will be developed by both hackers and the vulnerability testing community to exploit these type of systems. An important observation is that use of Windows-NT is not a panacea to providing network security. When configured properly, Windows-NT systems provide an effective security posture however, implementing the proper configuration is a technically challenging task.

Allocation of Resources:

The security posture of the OAN was obtained partly through the allocation of sufficient resources to secure this system. This is evidenced by an in-house, well-trained system administration staff section which effectively performed their jobs and associated security responsibilities. Rather than contracting out this function or using part time personnel who lack the expertise or motivation to perform this function, a work section of approximately 10 individuals perform this vital function. An extensive six month long process is used to fully train and orient newly assigned personnel to become qualified in the use of the diverse systems used in the OAN. In terms of physical facilities, the system administration function is largely performed in a single room, where all the OAN servers and gateways are located in close proximity to the system admin personnel on duty. A closely adjacent facility is used to prototype changes to the network configuration before changing the operational system. In addition, an auditing team separate from the system administrators also exists which effectively audits and monitors the OAN, as reported by the vulnerability testing team. The single auditor assigned to the OAN works closely with the system administrators, with the full time responsibility of reviewing system audit logs. This appeared to be a highly effective method of accomplishing this task since system admin personnel are usually focused on day to day activities with system fault conditions and implementing new users and capabilities.

The conclusion of this analysis was that the OAN is an effective model to demonstrate that even a system connected to the Internet can have an effective security posture. The OAN had several key factors working in its favor to achieve this rating. Obtaining this posture is not cost free or easy to accomplish. In addition to resources, it takes a strong level of commitment and management support to accomplish this feat.

Recommendations

While the OAN security posture was good, it was not perfect. Several recommendations were made to improve an effective security posture. The identification and authentication procedures for remote user E-mail access could be improved by using a security token to provide a one time session key, rather than the current practice of using static passwords. While the current E-mail guard implementation is resistant to intrusion, planning for this component's replacement should continue. The implementation of a security component "firewall" or guard directly incorporating the use of security tokens for a strong level of user authentication would represent a distinct improvement over the current E-mail guard. The commendable policy of having the OAN subject to "no notice" vulnerability testing should be continued. Since even a small change in a critical parameter, such as the router rules tables can have a drastic effect on the OAN's information systems security posture, it is essential that the OAN be subject to frequent vulnerability testing. Prudent risk management dictates that any system connected to the Internet should undergo periodic evaluations since it is subject to repeated intrusion attempts. Current security policies which limit risks by strictly controlling the use of modems, fax boards and off-network access should be continued and be enforced through random audits. The emphasis and resources placed on the system administration function are a key component of the OAN's security posture and should be continued. The innovative use of a separate auditing section to monitor security related events on the system was highly effective and should also be continued.

Conclusion

The increasing trend to network systems and connect these to the Internet has made securing current operational systems a high priority task. The intent of this case study is to show that it is possible to achieve an effective security posture with current technology, even on systems connected to the worldwide Internet. This type of security posture can only be achieved by allocating sufficient resources and providing the management commitment to ensure success. It is critical to note that even relatively small changes in a critical component, such as the router access control list, can have drastic effects on a system's security posture. The most effective method to determine the security posture of an operational system is to conduct periodic, in-depth vulnerability testing evaluations. The use of external intruder test teams in conjunction with the implementation of "no notice" vulnerability testing policies is a highly effective method to ascertaining the security posture of a networked system. The development and implementation of effective security policies are a critical first step to achieving a robust operational posture. Achieving an adequate security posture against both external and internal threats is a "do-able" task, and should be the objective of every networked system connected to the Internet.

References

1. Wack and Carnahan. U.S. Department of Commerce Special Publication 800-10. Subject: Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls, published by the National Institute of Standards and Technology (NIST), December 1994.
2. Cheswick and Bellovin. Firewalls and Internet Security, Addison-Wesley Publishing Company, 1994.
3. Mark Lottor, Subject: Internet Domain Survey January 1996. Source: <http://www.nw.com/zone/WWW/report.html>, 22 March 1996.
4. Briefing by Mr. Mike Higgins, then Chief DISA/Automated Information Systems Security Support Team (ASSIST), presented at the Multi Level Information Systems Security Initiative (MISSI) User's Conference, Orlando, Florida, October 1994.
5. Defense Information Systems Agency (DISA) document, Subject: The Electronic Intrusion Threat to National Security and Emergency Preparedness Telecommunications, published by the National Communications System (NCS), September 1993.
6. Briefing by Dr. Robert McKosky, Title: Security Technologies for the Information Superhighway, GTE Secure Systems Department, September 1994.

Internet Firewalls Policy Development and Technology Choices

Leonard J. D'Alotto
GTE Laboratories, Incorporated

Abstract

Since the development of the World Wide Web (WWW), more and more organizations are connecting their networks to the Internet. Many of these organizations are, rightly, concerned about the security of these connections. Realizing this, a number of companies are producing products known as Internet Firewalls and marketing them as the security solution. Faced with a blizzard of offerings, and a growing "feature war" between firewall vendors, confusion tends to be the order of the day. So what is to be done?

This paper addresses this question by arguing for the need for a proper Internet security policy. The information which should be included in that policy, and ways to use that policy for determining the appropriate firewall technology are given. This paper is based on the author's significant experience in evaluating firewall products and implementing them in a variety of environments.

1. Introduction

Since the development of the World Wide Web (WWW), more and more organizations are connecting their networks to the Internet. Many of these organizations are, rightly, concerned about the security of these connections. Realizing this, a number of companies are producing products known as Internet Firewalls and marketing them as the security solution. Faced with a blizzard of offerings, and a growing "feature war" between firewall vendors, confusion tends to be the order of the day. So what is to be done?

What must be realized is that firewall products are only a part of any Internet security solution, and the part they play differs not only between products, but between organizations and their Internet connections. These roles, the type of firewall product, and the need for features, depends upon the organization's policy towards Internet connectivity. That policy addresses not only security, but use of the Internet by employees and other issues related to the Internet. In addition, an organization must consider how that firewall product will be managed as part of the organizations overall network. This paper addresses the development of Internet policies, looks at available firewall technologies, and provides guidelines for applying the available technologies to meeting that policy.

2. Internet Connectivity Policy Development

Before an organization connects to the Internet, a policy governing that connection should be established. This policy should address three major areas: security, use, and management and administration. The organization must be aware, however, that policy development is a process that is not complete until all firewall and other technologies to be correctly implemented are chosen. In the process of implementing an Internet connection, an organization will usually tighten (or loosen) the policy based on risk mitigation vs. investment.

2.1 Security

This portion of the policy addresses what traffic is allowed to flow between the organization's network and the Internet. In setting these policies for an organization, more than just the security risk of a connection must be considered. For some organizations, inbound telnet is a real requirement. Inbound telnet is dangerous. Depending on the organizations mission and financial strength, strong user authentication and possibly encrypted sessions may be required. The key is to determine, for each potential Internet service:

1. Is there a real requirement to allow this service, both inbound and outbound,
2. What are the risks to the organization from this service,
3. What is the level of investment the organization is willing to make to mitigate these risks,
4. What are the preferred mitigation methods, and
5. What is the method for handling new Internet services?

Upon compiling this information, an organization can then proceed to develop the policy on who has access to Internet services.

The one item in this section that is frequently overlooked is the issue of how the organization will determine if new Internet services will be allowed. As new Internet services are developed, individuals within an organization will request access to these services. Since many of these services require 2-way communications, they could potentially cause an opening in the firewall. Therefore, a policy and set of procedures on how to request these services, and how to decide whether access to them will be allowed, must be developed. Otherwise, there will be no way of properly managing these requests and maintaining control over the firewall.

2.2 Use

This section of the policy addresses which members of the organization will be given Internet access, and the type of access they will be given. It is not unusual for an organization to give all employees the same access to the Internet. On the other hand, many organizations limit access to a "select" group of individuals, and then further restrict services within that group. For example, all professional staff may be given E-Mail access but only marketing and research (and

of course, all top executives) may be given access to the World Wide Web. The decisions made in developing this section of the policy are crucial to determining the type of firewall technology to be used.

The second part of this section involves determining the way in which access is to be restricted. Is it to be based on IP address, user ID and password, or via strong authentication of the user? All three options are available, but their impact needs to be understood. If access is to be limited based on the user's IP address, it is easily subverted. This can be done by any individual walking up to the authorized workstation, sitting down, and going to work. It can also be thwarted through IP spoofing and related techniques. The other problem with this is that every time a legitimate user moves, and thereby receives a new IP address, the rules in the firewall must change. With an account based restriction, where the user must authenticate their identity to the firewall, this administrative burden is removed as once an account is established, it is valid from all IP addresses within the enterprise.

2.3 Management and Administration

This section presents the guidelines for managing and administering the Internet connection. In setting these guidelines, there are several questions that must be answered. They include;

1. What events are to be logged,
2. Where are logs to be kept, and for how long
3. What events are to be alarmed,
4. What types of alarms are to be required, e.g., E-Mail, pager dialing, etc., and
5. What types of reports are to be required.

2.3 Remaining Issues

As one can see, by going through this process and documenting these policy decisions, the requirements for the firewall are almost complete. There are two major items missing, however. One is the skills and capabilities of those who will manage the firewall. If your organization is not staffed with capable Unix and TCP/IP administrators, this must be taken into account. One example is an old mainframe and SNA shop. While the people were bright and very professional, they were not expert Unix and TCP/IP administrators. Consequently, when one vendor's firewall was found to require the administrators to configure the system by editing Bourne shell scripts, it had to be rejected. This was in spite of the fact that at the time, it was one of the more robust firewalls available.

Another item to be addressed, and this is closely tied into the above, is who is to administer the firewall. Is this to be in-house, outsourced, or some combination of the two? All options are available in the market, and after careful consideration of the policy and the capabilities of available staff, a proper decision may be made.

Finally, the issue of compliance needs to be addressed. Unless a mechanism is put in place to ensure compliance, then this entire policy will be meaningless.

3. Firewall Technologies

What exactly is a firewall? A firewall is, simply, “A set of tools used to implement an Internet security policy” (heard by a participant in a late night discussion over beers at the November 1995 CSI conference). What this means is that the firewall product purchased from a vendor is not the entire firewall system. Rather, the firewall system includes the use, if applicable, of a protected mail host, a “split” Domain Name Service (DNS), and the use of a DMZ. While we give a brief introduction to these technologies here, the reader is referred to [2] for more information.

3.1 Proxy Servers and Packet Filters

These are the technologies encompassed in a typical firewall product. Simply put, a packet filter restricts based on source and destination IP address and TCP or UDP service port. A proxy server is “...a program that deals with external servers on behalf of internal clients. Proxy clients talk to proxy servers, which relay approved client requests on to real servers, and relay answers back to clients.” [2]. In practice, a proxy server requires users to log into the firewall, and then access the Internet from that server. There are now firewall products appearing on the market which perform packet filtering but require users to authenticate themselves. Once the authentication is complete, the firewall will allow traffic for that session to pass.

3.2 Use of a “DMZ”

A DMZ, a term stolen from [3], is simply a network between the firewall and the Internet over which you have some control. Figure 1 depicts such a network.

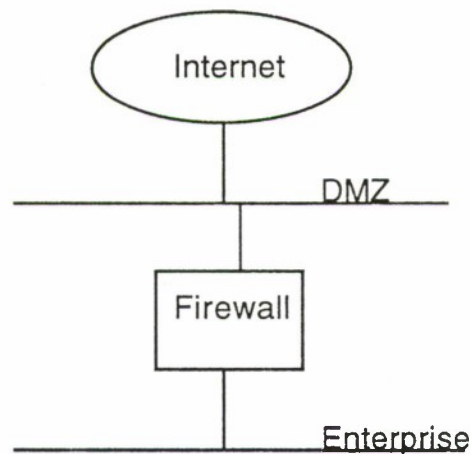


Figure 1 DMZ

The purpose of the DMZ is to have a place outside the firewall, but over which you still regain control, for external DNS servers, external mail hosts, public information servers, etc. This way, inbound traffic can be more tightly controlled, and information about the internal network kept from being published to the Internet. For more on DNS, Mail, WWW server, etc., security see [1], [2], [3], [4] and [5].

3.3 Encrypted Tunnels and Virtual Private Networks

The latest trend in firewall offerings is to add cryptographic services for firewall to firewall encryption. The encrypted traffic between these firewalls is referred to as an encrypted tunnel or a virtual private network (VPN). The purpose of this technology is to allow two sites connected to the Internet to use cryptography to communicate in secret and with total security. This then allows a company with multiple sites to use the Internet as a Wide Area Network (WAN) instead of paying for expensive private leased lines.

A word of caution is in order. If one wishes to use a VPN for this purpose, one must make sure the endpoints (firewalls or routers) are secure. The best cryptography in the world is easily subverted should the endpoints be easily penetrated and the cleartext visible to that penetrator.

4. Application of Technologies to Policy

The following case studies are based on actual situations. They are intended to illustrate the types of situations in which packet filtering, proxy servers, and packet filters with user authentication are appropriate.

4.1 Case Study - Packet Filtering

In this particular instance, the organization was currently connected to the Internet. The policy in place, and to be kept, was that all employees are to be given E-Mail, news, and outbound telnet, ftp, gopher, and http. Inbound services were to be limited to mail and news. In addition, DNS needed support, as did an anonymous ftp server and WWW home page. At the time, packet filters on the Internet router were being used to provide a rudimentary level of security. This was deemed inadequate as the rule base to implement a policy was complicated, and proper logging could not be supported on the router.

In this case, there was no need for authentication in either direction. Inbound traffic, being just E-Mail and News, could not be subject to any user authentication. Outbound was to be allowed, regardless of the workstation or user that originated that traffic. Additionally, all traffic could be restricted based on TCP or UDP service port, in conjunction with the source and destination address. (For example, an inbound connection on port 25 to the mailhost would be allowed, but to any other host in the organization, it would be disallowed.)

The result was a packet filtering firewall with a DMZ, split DNS, protected mail host, and appropriate event logging. Since internal user authentication was not required, as everyone had full Internet access, and no inbound traffic other than mail or news was allowed, no user authentication was required. With the split DNS and protected mail host, full service could be offered to users without publishing information as to the structure of the enterprise network. And, since most packet filtering firewalls provide fairly extensive logging facilities, the Internet security policy could be implemented for this organization fairly simply.

4.2 Case Study - Proxy Servers

In a second case, the policy was much more complicated than the first. The organization was extremely large, and people move offices fairly regularly. In addition, many users were on a LAN with a proprietary LAN OS which does not support native IP. This LAN also used a proprietary E-Mail transport with an SMTP gateway. The policy requirements resulted in five types of users - no access, E-Mail only, E-Mail plus news, full access restricted by day of the week and time of day, and unrestricted access. With the various types of access to be given to individuals within the organization, each request for outbound access would have to authenticate the user. With the constant changing of users workstation addresses, address authentication would be impossible to administer. Consequently, this was a classic case of where one uses a proxy server.

In addition, a robust DMZ for a protected mail host and support of a split DNS was required in this situation. The first reason for this is the E-Mail situation. With a proprietary internal mail system that utilized SMTP gateways, there needed to be some way of managing traffic to those gateways. An external mail host was used to simply relay mail to one of several internal mail hosts, on which an SMTP gateway resided. This reduced the processing load on the firewall, as it was not required to determine which mail host a message should be relayed to. E-Mail addresses were set up to be of the form `user@mailhost.domain`, and the external mailhost was configured to only know the addresses of the internal mailhosts. Coordinating this with a split-DNS allowed for separation of the Internet and the LAN-OS based users, as well as protecting against publicizing the structure of the IP based portion of the network.

4.3 Case Study - Filters with User Authentication

Where does one use a packet filter with user authentication? As it so happens, this was not done in linking to the Internet, but rather, in linking contractor and vendor networks to the corporate LAN. In this particular case, a variety of contractors and vendors need access to the organization's LAN, but only to limited machines. The contractors and vendors all came from registered Class B or Class C networks. However, traffic flow from a contractor needs to be limited to those machines to which they are authorized access. This allows for a simple rule base where the rules are of the form:

From	To	Service	Action
Class B or C address	Host Addresses	Service Ports	Allow

The problem is, not all personnel from the contractor are allowed access. Therefore, user authentication is needed. Now, the question arises, why not just use a proxy server? The answer is that the contractors are doing development and maintenance and need a variety of services for which there are no available proxy servers. So, setting up a packet filter with user authentication presented itself as the only alternative. That is, if the traffic matches an allow rule, authenticate the user as one authorized to generate this traffic, and pass it. Otherwise, drop it.

5. Summation

In closing, one can see that preparing a robust policy is a prerequisite before making a choice on how to firewall an Internet connection. If such a policy is written, and the approach to implementing the firewall chosen and documented, these can then be given to firewall vendors. The vendors should then be asked to provide, in writing, a document describing how they will implement your policy in their product. Upon receiving these documents, choose two or three finalists, and have them provide evaluation systems. Place these systems in a laboratory in which the real connection can be simulated. Extensively test these products before making a decision. In this way you can gain a proper understanding of how the firewall product will fit into your organization. From this, you can choose a vendor and implement your firewall.

6. References

[1] Paul Albitz, Cricket Liu; DNS and Bind; ; O'Reilly & Associates © 1992

[D. Brent Chapman and Elizabeth D. Zwicky; Building Internet Firewalls, O'Reilly & Associates © 1995

[3] William R. Cheswick, Steven M. Bellovin; Firewalls and Internet Security - Repelling the Wily Hacker; Addison-Wesley Professional Computing Series, Addison-Wesley Publishing Company, © 1994

[4] Bryan Costales, Eric Allman, Neil Rickert; sendmail; O'Reilly & Associates © 1993

[5] Cricket Liu, Jerry Peek, Russ Jones, Bryan Buus, Adrian Nye; Managing Internet Information Services; O'Reilly & Associates © 1994

A Case for Avoiding Security-Enhanced HTTP Tools to Improve Security for Web-Based Applications

by Bradley J. Wood[†]

Abstract

This paper describes some of the general weaknesses of the current popular Hypertext Transmission Protocol (HTTP) security standards and products in an effort to show that these standards are not appealing for many applications. We will then show how we can treat HTTP browsers and servers as untrusted elements in our network so that we can rely on other mechanisms to achieve better overall security than can be attained through today's security-enhanced HTTP tools.

Introduction

The World Wide Web (WWW or the Web) has become the new popular computing paradigm for applications developers and decision makers. It is becoming increasingly popular to develop new applications and networks based on this model. There is also a lot of interest in migrating legacy applications to a Web-based infrastructure.

Unfortunately, we are finding that HTTP is not well suited to applications that have even the most basic security requirements. This limits the usefulness of the Web to applications that have few real security requirements. More importantly, decision makers are opting to develop Web-based applications to gain increased functionality at an admitted loss of security and control. Therefore, it is important that we develop the tools and techniques needed to satisfy our basic security requirements in a Web-based infrastructure.

A lot has been written in even the popular press about coming advances that promise secure WWW applications. Unfortunately, most of the current and emerging products and standards for adding security to HTTP are lacking; and, it is not clear that we will see satisfactory advancements in the foreseeable future. Therefore, applications developers are left to their own devices to create security-enhanced HTTP applications using currently available techniques and technologies.

In this discussion, we will examine some of the general weaknesses in the current security-enhanced HTTP products and standards. As an alternative, we will review techniques for satisfying security requirements without using any of the HTTP security enhancements.

Current Standards and Products

There has been a lot of activity recently in the area of security products and standards for the WWW. Two different standards are evolving as the dominant choices for adding HTTP security: the Secure Sockets Layer (SSL), and Secure HTTP (S-HTTP). It has also been reported that there are efforts underway to integrate SSL and S-HTTP into a universally accepted Web security solution[1]. Unfortunately, today's reality is quite different than some of the promised results.

One could argue that Web developers are faced with a difficult choice for adding standards-based security to their Web-based applications. One solution, epitomized by the *Netscape Navigator* and *Netscape Commerce Server*, provides relatively little security in favor of improved overall application robustness and a rich set of features. The other standard, S-HTTP, offers a robust set of security features on browsers and servers that are generally not as robust or full-featured as the Netscape product family.

[†] Bradley J. Wood is a Senior Member of Technical Staff in the Data Systems Security Department of Sandia National Laboratories, Albuquerque, NM 87185-0451. He can be reached by phone or fax at (505) 845-8461 or by electronic mail as Brad.Wood@Sandia.gov. A revised copy of this paper will be maintained on the World Wide Web at <ftp://saix1130.endo.sandia.gov/NISSC.html>

Secure Sockets Layer

SSL[2] was originally developed by Netscape Communications [3] to enhance a Web browser and server to reduce the risks of exchanging sensitive information. The primary application for SSL is to allow a consumer to use a Web browser to purchase products and services using a credit card number for payment information. In this model, it is important to protect client information (like the credit card number) during the transaction. SSL is currently implemented in the *Netscape Commerce Server* [4] and the *Netscape Navigator* browser, as well as other products.

The primary security service offered by the *Netscape Commerce Server* is to establish a private (encrypted) communications channel between a server and a browser. This allows strangers to exchange information privately. This is useful if you are a merchant collecting orders from a variety of buyers on the Web, but it appears to have few other applications.

The *Netscape Commerce Server* also provides a relatively-strong mechanism for authenticating servers to browser users, provided the client checks the server certificate when a secure session is established, and provided that the server certificate is genuine. Client authentication is currently limited to a username / password technique.

Although there are relatively few advanced security features in the *Netscape Commerce Server*, there is still a lot of interest in using the Netscape product family. Some of the perceived Netscape advantages include:

- **Simple Key Management** - Server certificates are validated using public signature keys that are embedded in the *Netscape Navigator*. Browser users are not required to do anything to enable the SSL features in the browser. Therefore, every Netscape browser comes with these basic security features already enabled and ready to use.
- **Rich Feature Set** - The *Netscape Commerce Server* has an applications programming interface (API) and other features that allow content providers to create attractive and feature-rich Web sites. In a competitive environment, content providers are eager to leverage any feature that will distinguish their service among the multitude of sites on the Web.
- **Widely Distributed Browser** - The *Netscape Navigator* has been distributed as shareware, so it is readily available to anyone with even casual access to the Internet. Still, this browser is widely

touted as being one of the most stable and feature-rich Web browsers in the industry. There are versions of the *Netscape Navigator* available for most major computing platforms including *Microsoft Windows*, Apple's *Macintosh*, and X-Windows under many different versions of UNIX. As a result, many industry sources report that the *Navigator* is the dominate Web browser on the market.

We are seeing a lot of interest in modifying the *Netscape Commerce Server* and *Netscape Navigator* to provide strong authentication of the browser user to the server. Some of these enhancements leverage Kerberos, DCE, and one-time password technologies.

We are also seeing a distressing number of successful attacks against Netscape's implementation of SSL and other security features in both the server and the browser [5]. This appears to be a logical result of the enormous pressure that the market has placed on Netscape to add features to their products as quickly as possible. Although Netscape has entered into an agreement with RSA Data Security to review their security implementations in the future, it is not clear that the market will ever demand fastidious security implementations at the expense of longer product or feature development cycle times.

Secure HTTP

Secure-HTTP [6] (S-HTTP) is the other major standard proposed for Web-based security enhancements. S-HTTP was originally developed by a team at CommerceNet and Enterprise Integration Technologies (EIT) [7] to provide a robust set of security services for a variety of applications, particularly robust commercial electronic commerce over the Internet using a Web-based infrastructure.

The primary strength of the S-HTTP specification is that it characterizes a rich set of robust, negotiable security features. S-HTTP has the potential to satisfy a variety of security requirements for both clients and servers using sophisticated cryptographic techniques. Indeed, S-HTTP could potentially solve most common Web-based security requirements.

Unfortunately, S-HTTP is not as widely deployed as SSL. Although we have had tool kits and prototype implementations for some time, there are relatively few production-quality products and applications using S-HTTP, and the S-HTTP community appears to be evolving more slowly than other product families (such as Netscape's).

In addition, the security features in an S-HTTP application must be fastidiously designed and implemented.

There is often a complicated client enrollment process that must be performed in advance of establishing an S-HTTP session between a browser and a server. Most of these enrollment processes involve cryptographic key management and registration tasks.

We are also somewhat distressed by the poor quality of some of the products that implement S-HTTP. Many S-HTTP browsers and servers are built upon shareware or public domain products that themselves have some significant security problems. We have also noticed that most of the browsers that implement S-HTTP do not offer the features and overall robustness of the *Netscape Navigator*. There appears to be relatively little interest in widespread adoption of any S-HTTP browser in favor of the Netscape browser.

Other Approaches

We have also seen other approaches for adding security to a Web-based infrastructure that are not widely implemented but often mentioned in the some of the popular literature.

DCE - Proposals have been made to process standard HTTP transactions over an infrastructure that uses Distributed Computing Environment (DCE) security services [8]. Here, client workstations and servers use DCE security services to establish a trusted session or channel where standard HTTP transactions are supported. Although this approach requires that the user invest in a relatively-expensive DCE infrastructure, this approach may be appeal to enterprises that have already invested in DCE and who only need security enhancements for applications that run over their current DCE infrastructure. Another advantage of this approach is that you can use robust DCE security services without major modifications to the HTTP browser or server.

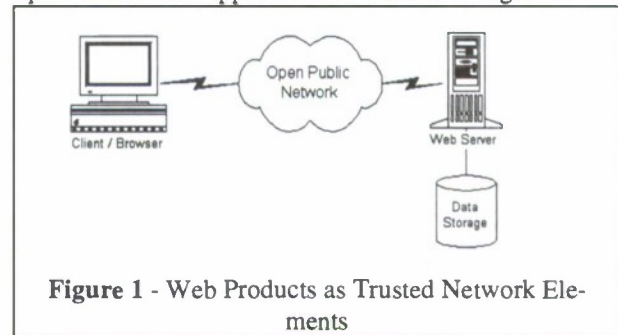
Kerberos - Enterprises that already use Kerberos security services are seeking to leverage that investment to improve their Web-based applications. We have seen some work at facilities like Sandia National Laboratories (New Mexico) where Kerberos is being integrated into the *Netscape Commerce Server* to provide strong user authentication. This approach uses an unmodified Netscape browser to securely pass a Kerberos username and password to the *Commerce Server* using SSL. The server then performs the Kerberos initialization function to verify the identity of the browser user and to obtain the access privileges (or tickets) for that user.

An Alternative Approach

An alternative approach to satisfying security requirements in a Web-based application is to simply treat the HTTP browser and server as untrusted elements in the computing network. We will introduce this approach by contrasting it with the approach that relies on satisfying their security requirements using Web-based products or services.

Web Tools As Trusted Elements

In this approach, we want to rely on features in the Web browser and server to satisfy our security requirements. This approach is illustrated in Figure 1.



This approach is fairly common, and it is characterized by the following features:

- We rely on the Web browser and server to cooperatively authenticate each other and determine the identity of the browser user or client for the server.
- We rely on the Web browser and server to cooperatively protect the data exchanged over the open public network.
- We rely on the Web server software or operating system to enforce access controls to the stored data base.

This approach is popular, primarily because this model can be developed with a minimum investment in hardware, software, planning, and training. It leverages the advertised security features of the Web-based products.

Unfortunately, there are several potential problems with this approach:

- Server Processes with Vulnerabilities** - The common expectation is that if you install a Web server software package on a respectable computer, you will have a full-feature production Web site. In reality, to get most of the desired features, administrators must install a variety of network

server processes on their computer. Some of these extra required server processes might provide file transfer, electronic mail, or database management services. Although this is technologically feasible, each server process has its own potential security weaknesses that an adversary could exploit to gain unauthorized access to the stored data. Therefore, the more products that we install on a single server, the more likely that server will become vulnerable or compromised.

- b) **Access Controls** - It is not clear that we can trust the Web server software to provide fastidious access controls to the stored data. What we are seeing is that the current Web servers provide either few access control features, or they rely completely on the server operating system for data access control. Therefore, this approach may not be suitable for applications that need rigorous access control.
- c) **Weak Authentication** - We are seeing many implementations that rely simply on traditional username / password pairs for authenticating parties. This is widely regarded as a weak technique. However, most Web servers do not offer stronger or more sophisticated authentication services. In addition, it is difficult to do rigorous access control based on weak authentication.
- d) **Catastrophic Compromise** - In the likelihood that the single server is compromised through any number of common attacks, the entire information system is compromised. This event could be catastrophic if the stored data is sensitive in any way.
- e) **Exposure** - Web servers are generally installed outside a traditional firewall or other security gateway. This is necessary because most Web functions are hindered by a firewall, proxy server, or other technique; and, the primary requirement for most Web servers is availability. Unfortunately, this makes the server highly vulnerable to a variety of potentially sophisticated adversaries.

Web Tools As Untrusted Elements

In the alternative approach, we treat the Web products as untrusted computing elements; and, we do not rely on these products to enforce our security policy. An example of this approach is shown in Figure 2.

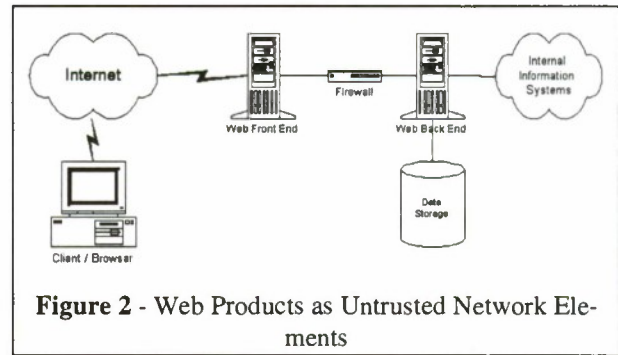


Figure 2 - Web Products as Untrusted Network Elements

This approach is characterized by the following features:

- There is a clear boundary between internal information systems or servers and external resources. The boundary is typically a firewall, proxy server, or some technique to limit the exposure of the internal network.
- The external Web server or Front End is used simply as a user interface. Most of the actual information processing is done on resources in the internal network.
- Access controls and other security requirements are usually satisfied by using database management systems or other products with rigorous access controls.
- Authentication is done between the actual browser user (the client) and internal information systems. The external Web Front End does not participate in the actual authentication process.

There are some distinct advantages to this technical approach:

- a) Since the Web Front End is used as strictly a user interface, you can expect better performance than a system that must support many server process. In addition, this Web Front End can be optimized for its unique role.
- b) This approach gives the network designer the ability to integrate a variety of well understood or mature security techniques into a Web-based infrastructure, such as security-enhanced messaging.
- c) This technique also allows the designer to integrate traditional or legacy information systems such as database management systems into a Web-based infrastructure.

Unfortunately, this approach has one primary weakness. This approach is generally more complex than the traditional approach, leading to increased expense to procure and manage. This approach also requires

that the information system be designed by experienced information systems security professionals.

Applications-Layer Security Protocols

We are also seeing a lot of activity in the WWW community on applications-layer security protocols. These protocols are really designed to work on top of or independent of a particular Web browser or server. Examples of these protocols include:

- Secure Transaction Technology (STT) developed by Visa and Microsoft [9]
- *Secure Courier* developed by MasterCard and Netscape [10]
- Secure Electronic Payment Protocol (SEPP) developed by IBM and others [11]

These protocols [12] all provide an independent means for developing strong security techniques at the Applications Layer. Therefore, these protocols could be added to Web browsers, servers, back end systems, and electronic messaging systems.

The point is that the WWW community has identified the benefit of moving their security mechanisms outside the WWW products, and this is just another approach to satisfying the security requirements for a family of applications.

New Developments

The market for advanced HTTP products is responding with new products at an amazing pace. Users are demanding improved HTTP security, and some vendors are responding with announcements of improved security features in their future products. For example, Netscape Communications has made several new product announcements.

- Netscape announced that a new version of their browser, *the Netscape Navigator v2.0*, will be generally available in January 1996 [13]. One intriguing feature of this product is the addition of a client-side digital certificate for public key applications. However, it is not clear how a user would actually take advantage of this digital certificate. We speculate that the primary purpose of this certificate is to support the security-enhanced messaging features that have also been added to the v2.0 browser. Ideally, we would like to use this certificate to provide a strong client-side authentication to the *Netscape Commerce Server*. However, it is not clear that this will be supported in the v2.0 browser.

- Netscape has also announced that they plan to release a new version of the *Netscape Commerce Server* in the first quarter of the 1996 calendar year [14]. Unfortunately, we have seen no information on what security enhancements might be in this server.
- Netscape has also announced that they plan to develop a family of security-enhanced Web products that incorporate the National Security Agency's FORTEZZA technology [15]. Current plans call for this product to be available sometime in late 1996, and it is not clear what features will actually be supported in any of the components.

Summary

There is a great deal of interest in adding security features to HTTP- or Web-based applications. Unfortunately, it is not clear that we can satisfy even our most basic security requirements with current security-enhanced HTTP products. There is no indication that planned product enhancements will fully rectify this situation.

Therefore, it is up to the applications developers to satisfy their security requirements using current technology. One technique that appears to satisfy this goal is to treat the WWW components in a network as untrusted elements, and use traditional techniques to enforce the security policy. This approach leads to networks that can be complex and expensive, but it appears to be the only way to implement a reasonable security policy on a Web-based infrastructure.

References

The majority of these references are Uniform Resource Listings (URL)s to hypertext documents on the World Wide Web.

- [1] Press release on Terisa Systems Partnership: <http://dengue.terisa.com:80/new/pr/041095b.html>
- [2] References on Secure Sockets Layer: <http://home.netscape.com/newsref/pr/newsrelease17.html>,
<http://home.netscape.com/newsref/std/sslref.html>
- [3] Netscape's home page on the Word Wide Web: <http://home.netscape.com>
- [4] *Netscape Commerce Server Reference Guide* and *Netscape Commerce Server Programming Guide*, 1995, Netscape Communications, Inc.
- [5] References on attacks and vulnerabilities in Netscape products: <http://home.netscape.com/newsref/pr/newsrelease68.html>, <http://home.netscape.com/newsref/pr/newsrelease46.html>, http://home.netscape.com/newsref/std/random_seed_security.html; <http://www.openmarket.com/press/nssecurity.html>
- [6] Secure HTTP specification: <http://www.eit.com:80/creations/s-http/>; <ftp://ds.internic.net/internet-drafts/draft-ietf-wts-shhttp-01.txt>
- [7] Enterprise Integration Technologies home page on the World Wide Web: <http://www.eit.com/>
- [8] Information on the Open Systems Foundation's DCE-Web project: <http://www.osf.org/www/dceweb/index.html>
- [9] References on the Secure Transaction Technology (STT) specification: <http://www.visa.com/cgi-bin/vee/sf/commerce/sttdownloads.html?2+0>, <http://www.windows.microsoft.com/windows/ie/stt.htm>
- [10] References to the Secure Courier specification: <http://home.netscape.com/newsref/pr/newsrelease33.html>,
<http://home.netscape.com/newsref/pr/newsrelease10.html>, <http://home.netscape.com/newsref/std/credit.html>
- [11] Reference on the Secure Electronic Payment Protocol: <http://www.mastercard.com/Sepp/sepptoc.htm>
- [12] Announcement of intent to merge payment standards: <http://www.mastercard.com/Press/release-960201.htm>
- [13] Information on Netscape Navigator v2.0 features: <http://home.netscape.com/newsref/pr/newsrelease43.html>,
<http://home.netscape.com/newsref/pr/newsrelease82.html>
- [14] Information on new version of the Netscape Commerce Server: <http://home.netscape.com/newsref/pr/newsrelease43.html>
- [15] Netscape FORTEZZA announcement: <http://home.netscape.com/newsref/pr/newsrelease49.html>



Security Mechanisms for the World Wide Web

Bradley J. Wood
(brad.wood@sandia.gov)

Sandia National Laboratories

Albuquerque, New Mexico 87185-0451
Phone & Fax: (505) 845-8461

6/17/96

----- Preliminary -----

Page 1



Outline

- Some Basics
- Industry Standards
 - Netscape's Approach
 - Others
- Non-traditional Approaches
 - "Untrusted Server"
 - Kerberos, DCE, others
- Future Developments

6/17/96

----- Preliminary -----

Page 2



Some Basics

- Know your real requirements.
- These products are for unclassified use only.
 - Not trusted to separate red & black data
 - May be used with red data on a system-high network
 - May address need-to-know problems

6/17/96

----- Preliminary -----

Page 3



More Basics

- A security-enhanced Web server is vulnerable to other attacks.



6/17/96

----- Preliminary -----

Page 4



Industry Standards

- The Secure Sockets Layer (SSL) is dominant
 - Invented by Netscape in 1994
- Security services:
 - Simple key management
 - Privacy among complete strangers
 - Authentication of server to client (limited)
 - Traditional client authentication (username / password)
 - Access controls at server directory level

6/17/96

--- Preliminary ---

Page 5



Other Approaches

- Kerberos Authentication
 - Modifications server to do Kerberos authentication
 - Requires Kerberos infrastructure
 - Implemented at Sandia New / Mexico
- OSF/DCE
 - Requires DCE-aware clients and servers
 - Requires DCE infrastructure

6/17/96

--- Preliminary ---

Page 7



Other Industry Standards

- Secure HTTP (SHTTP)
 - Invented early 1995 by EIT, Inc. for CommerceNet
 - Security added through additions to the HTTP files
 - Robust security services
 - Relatively complex to design and develop
 - Limited availability, especially for Windows NT

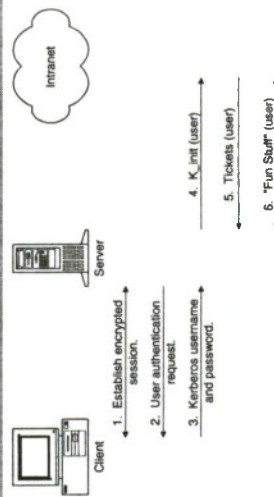
6/17/96

--- Preliminary ---

Page 6



Kerberos Authentication



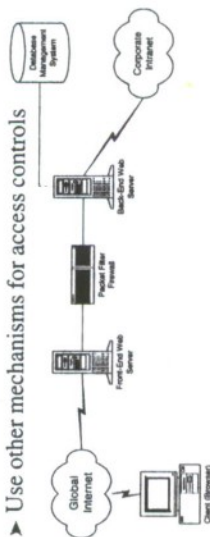
6/17/96

--- Preliminary ---

Page 8

"Untrusted Server" Approach

- Assume the Web server will be compromised
- Keep all of data behind another firewall
- Use other mechanisms for access controls



8/17/96

--- Preliminary ---

Page 9

Promised Enhancements

- Netscape Navigator v3.x
 - Certificate-based 2-way authentication + S/MIME
- FORTEZZA-based Netscape Products
 - 2-way authentication & encrypted objects
- Applications-layer Security Protocols
 - Secure Electronic Transactions (SET) among others

8/17/96

--- Preliminary ---

Page 10

Summary

- Traditional security requirements don't fit the Web paradigm.
- Designers have some difficult standards choices:
 - SSL vs. SHTTP
- Non-traditional approaches hope today.
 - Kerberos, DCE, and others
 - "Untrusted Server"
- Future enhancements also offer hope.

8/17/96

--- Preliminary ---

Page 11

For More Information

- A copy of this talk and paper is available on the Web!
 - ftp://saix1130.endo.sandia.gov/NISSC.html

8/17/96

--- Preliminary ---

Page 12

APPLYING THE EIGHT-STAGE RISK ASSESSMENT METHODOLOGY TO FIREWALLS

David L. Drake (david_drake@cpqm.saic.com)
Katherine L. Morse (katherine_morse@cpqm.saic.com)
Science Applications International Corporation
10770 Wateridge Circle
San Diego, CA 92121
© 1996 SAIC

ABSTRACT

The explosive growth of the Internet has brought thousands of companies exciting, new electronic contact with their customers. It has also brought them equally exciting contact with a cadre of ingenious and persistent hackers. Increasingly companies are turning to firewalls to thwart these wily hackers. While firewalls are very effective, often they are not the security panacea they are made out to be. This paper presents a risk assessment of a hypothetical firewall using the Security-Specific Eight-Stage Risk Assessment Methodology which illuminates where the security flaws lie. The example serves as guidance for assessing firewalls in general. We discuss the lessons we learned performing actual assessments which lead to recommendations for improving the security surrounding firewalls.

INTRODUCTION

In our 1994 paper [1] we identified three major flaws in existing security risk assessment methodologies. We presented our new security-specific eight-stage risk assessment which addresses these shortcomings. In this paper we show how the methodology may be applied to a firewall, a security mechanism of considerable current interest. We begin with a brief overview of the methodology. The overview is followed by a representative application of the methodology to a firewall that was drawn from our firewall evaluations. The results of the assessment lead us around to a recurring security risk and a proposal for improving firewalls to address this risk.

1. THE EIGHT-STAGE METHODOLOGY

This risk assessment of a generic, hypothetical firewall employs the Security-Specific Eight-Stage Risk Assessment Methodology [1]; henceforth referred to as the eight-stage methodology. The eight stages of the methodology are illustrated in Figure 1, The Eight-Stage Model.

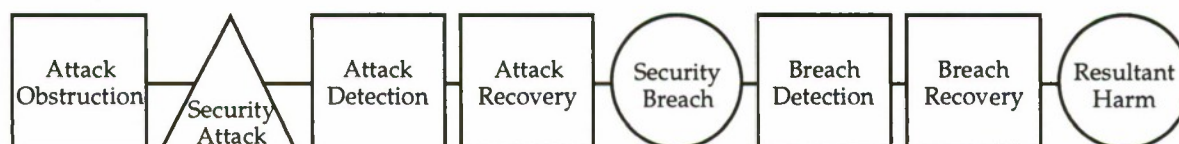


Figure 1. The Eight-Stage Model

In Figure 1, time flows from left to right. The internal influences are depicted as squares. The external influence (a security-related attack) to the system is depicted as a triangle. The consequences are depicted as circles. The objective of the security system is to prevent unwanted consequences of the security attack by employing the activities represented in the squares. The consequences, represented by circles, will occur if these activities are insufficient. One of the major principles of the model is that a system under attack has three opportunities to reduce the resultant harm: before the attack occurs, after the attack occurs but before a security breach occurs, and after a security breach occurs but before the resultant harm occurs.

When performing an assessment, we assess more than the firewall itself. We include both the automated security mechanisms of the firewall and the procedural requirements levied on the users and administrators. We refer to this without ambiguity as the "system". The eight-stage model is used to evaluate this system.

Performing an assessment using the eight-stage methodology involves two major steps:

- data gathering
- construction of eight-stage chains of security-relevant events and performing the quantitative analysis.

Both of these steps are described in the subsections below.

1.1 Gathering Data

The steps to gather the data for the assessment are:

1. Obtain the definition of the security boundary and the interfaces that will be defended by the firewall, both automatically and procedurally. The definition should be provided in the security policy.
2. Obtain the list of system assets to be protected, what constitutes a security breach, the associated harm that could befall the assets, and a quantitative loss per asset if it were compromised, modified by an unauthorized agent, or its availability were lost. This list should also be provided in the security policy.
3. Delineate the attack scenarios that will (and will not) be defended against, and the likelihood of occurrence of each. For firewall assessments, we have collected a long list of attack scenarios that cover most insider and outsider attacks.
4. Delineate each of the system's countermeasures that protect it against attack. A determination is made for each countermeasure if it is used to obstruct, detect or recover from an attack, or to detect or recover from a security breach. This distinction is used to support the quantitative assessment of each countermeasure's effectiveness.

1.2 Constructing the Chains and Performing the Analysis

The lists resulting from the data gathering phase are used to construct eight-stage event chains. One eight-stage chain is constructed for each attack scenario. In the appropriate stages, all applicable countermeasures, breaches, and are listed.

For each of the attack scenarios, the system's ability to defend against it is calculated based on the quantitative measures collected in section 1.1. There are eight data points that are collected during the data gathering phase for the eight-stage event chains: the effectiveness of the attack obstruction (CE_{AO}), the likelihood of an attack within one year (PR_A), the effectiveness of the attack detection (CE_{AD}), the effectiveness of the attack recovery (CE_{AR}), the loss in dollars if a security breach occurs (PL_B), the effectiveness of the breach detection (CE_{BD}), the effectiveness of the breach recovery (CE_{BR}), and the total value in dollars of the assets at risk in the attack scenario (PL_H). The likelihood of an attack is stated as the average number of attacks that will occur within a year. This is a departure from our 1994 paper [1] where we had assumed that the number of attacks would be less one per year. Anyone who reads the newspaper knows that this is no longer the case. All effectiveness measures are stated as probabilities ranging from 0.0 to 1.0. In our years of using this methodology, we found that the actual loss in dollars related to the security breach, PL_B , was always so low that it was not worth tracking. For example, if a hacker were breaking into a system, most security policies state that a security breach has occurred as soon as the hacker, as an unauthorized user, has somehow logged into the system. But at that point, no real dollar loss has occurred. There are exceptions, but to simplify the equations in this paper, we will assume that PL_B is always zero.

The likelihood that an attack will happen in one year and successfully result in a security breach (ER_B) is $PR_A \cdot (1 - (CE_{TO} + ((1 - CE_{TO}) \cdot CE_{AD} \cdot CE_{AR})))$. The likelihood that an attack will happen and successfully result in a harm (ER_H) is $ER_B \cdot (1 - (CE_{BD} \cdot CE_{BR}))$. The potential dollar loss per year (EL_T) for the one attack scenario being analyzed is $PL_H \cdot ER_T$.

It is important to keep in mind that the effectiveness measure for the attack obstruction (CE_{AO}) is the combined effectiveness of all of the mechanisms being used for attack obstruction against the one attack scenario being analyzed. Additional analysis may need to be performed to determine how all of these attack obstruction mechanisms interplay. The same is true for the analysis of the effectiveness for all detection and recovery mechanisms. In cases where mechanisms have different reactions based on situations, it may be necessary to decompose the analysis into more specific attack scenarios, resulting in more eight-stage event chains to analyze. See our earlier work [1] for the underlying formulation of all calculations.

2. APPLYING THE METHODOLOGY TO AN EXAMPLE FIREWALL

The example firewall that we will use is an amalgamation of the actual systems that we have assessed. The asset values, likelihoods, and effectiveness measures used in the example are drawn from these assessments. Our example firewall is a bastion host using IP-based filtering with an external router connected to the Internet. It is used to protect company proprietary data, including financial and Privacy Act data, on a collection of LANs supporting various computing platforms. We constructed this example system because of its commonality to current firewall installations. Our example allows only the following data flows:

- e-mail in both directions
- both internal and external hosts are allowed to "ping" the firewall (for connectivity testing)
- both in-coming and out-going Domain Name Service (DNS) requests
- non-anonymous File Transfer Protocol (ftp)
- World Wide Web.

In the following subsections we proceed through an abbreviated assessment, following the steps described in subsections 1.1 and 1.2. Given the space constraints, the tables provide examples and are not exhaustive.

2.1 Gathering Data

Table 1, Security Policy, synthesizes the example firewall's security policy. While the security policy should be provided by the system owner, in all of our assessments that was not the case and developing the security policy was our first task. The table is divided into three sections: the security boundary, the automated defenses of the firewall, and procedural defenses which are the responsibility of the users and administrators.

An abbreviated list of the assets to be protected are given in Table 2, Protected Assets. Listed with each asset are the types of breaches associated with its loss, the type of the resulting harm, and the value of the resulting harm. Table 3, Attack Scenarios, lists some of the attack scenarios that will and will not be defended against by the firewall, and the likelihood of occurrence of each. The attack scenarios that will not be defended against are addressed so that a true level of vulnerability can be assessed. Impossible attack scenarios for this example, such as those using telnet, are excluded since the firewall completely precludes their being enacted.

Table 1. Security Policy

Security Boundary
All internal network nodes and the firewall itself
Automated Defenses
Users on the outside network and users on the inside network are prohibited from all interaction with the firewall with the exception of e-mail, ping/echo, DNS, and an extremely limited ftp capability.
E-mail is allowed to pass between the internal network and the Internet.
Users on the external network are allowed to ping the firewall.
DNS is allowed for both in-coming and out-going requests and replies.
Outbound requests for file transfers using ftp from the internal network to the Internet are permitted.
Inbound requests for file transfers using ftp from the Internet to a designated ftp site within the internal network are permitted.
Outbound requests from the internal network for WWW access to the Internet are permitted, with Java disabled.
Internal network addresses are hidden from the external network.
Procedural Defenses
Users are not allowed to modify the e-mail program.
Users are not allowed to e-mail proprietary and/or private data over the Internet.
Users are not allowed to automatically forward e-mail to the Internet.
Administrators of the firewall must securely administer the system.
Users must be wary of all data received over the Internet, independent of its source.
Users and administrators must take great care in selecting programs which support web browsers.
Proprietary or private data must never be placed in the outgoing ftp directory.

Table 2. Protected Assets

Asset	Breach*	Harm†	Value
Firewall CPU time	A	R, T	\$100/hr.
Firewall system files	I	M	\$1,000/file
Firewall disk space	A	R	\$300/Mb
Web site on firewall	I, A	R, T	\$400
Firewall password file	C, I	M	\$1,000
Ftp file site	A	R, T	\$2,000
Firewall e-mail service	A	R, T	\$500
CPU time on non-firewall systems	A	R	\$500
Privacy Act Data	C, I, A	M, P	\$10,000
E-mail messages	C, I	M	\$5000
Financial records	C, I, A	M, D	\$50,000

*C = loss of confidentiality, I = loss of integrity, A = loss of availability

†M = failure of mission, P = loss of personnel, R = loss of resources, D = loss of dollars, T = loss of time

Table 3. Attack Scenarios

Attack Scenario	Defended Against	Likelihood
Hacker floods firewall network ports	No	.01
Hacker peruses e-mail traffic	Via procedures	.01
Hacker forges e-mail return address	No	5.00
Hacker attempts to use the sendmail security holes	Yes	2.00
Hacker spoofs Internet's DNS	Yes	.01
Hacker attack on FTP	Yes	6.00
Viruses received via the WWW infect internal programs	Via procedures	3.00
User inadvertently violates security policy	Via procedures	100.00
System administrator inadvertently misconfigures firewall	Via procedures	3.00

Table 4, System Countermeasures, lists several of the countermeasures that the system provides and their types.

Table 4. System Countermeasures

System Countermeasure	Type
Packet blocking	Obstruction
Packet filtering	Obstruction
Services written with secure features	Obstruction
Security education	Obstruction
Audit log analysis	Attack & Breach Detection
Automated alarms	Attack & Breach Detection
User detection of file modification	Breach Detection
User detection of mail spoofing	Attack Detection
Statistics utility results analysis	Attack & Breach Detection
User detection of system malfunction	Breach Detection
Firewall reconfiguration	Attack & Breach Detection
Firewall shutdown	Attack & Breach Detection
Firewall reinitialization	Attack & Breach Detection
Turning off firewall services	Attack & Breach Detection

2.2 Constructing the Chains and Performing the Analysis

Since space does not permit reproducing the results of all attack scenarios, we have selected two representative samples. A typical assessment would have approximately 80 chains. The first example, Table 5, Automated Attack Scenario, illustrates an attack against which the firewall is designed to protect. The second, Table 6, Human Error Scenario, illustrates the type of human error against which the firewall cannot protect itself.

The scenarios are presented in tables, each containing an eight-stage model of the attack being enacted. The tables should be read as a time-line, progressing from stage 1 through stage 8. The stages are listed in the first column, and the instance in the second column. The third column provides the quantitative measures associated

with the instance as described in section 1.2. The important results are the total effective risk, ER_T , and the loss that is associated with it, EL_T . Table 5 is the analysis of a hacker attacking a firewall protocol that is allowed, sendmail's SMTP protocol, but is secured by the use of the latest version of sendmail. It's obstruction effectiveness is very high, but it's not guaranteed to be impenetrable. This is reflected in the bottom line by the low level of risk and effective loss.

Table 5. Automated Attack Scenario: sendmail attack

Stage	Instance	Effectiveness, likelihood, or potential loss level
1. Attack obstruction	Service written with secure feature: firewall's use of secure version of sendmail.	Effectiveness (CE_{AO}): .99
2. Attack scenario	Hacker attempts to use the sendmail security holes to gain access to firewall.	Likelihood (PR_A): 2.0
3. Attack detection	Audit log analysis; automated alarms	Effectiveness (CE_{AD}): .9
4. Attack recovery	Turning off firewall services; firewall shutdown	Effectiveness (CE_{AR}): .9
5. Security breach	Hacker gains access to firewall CPU time, system files, and disk space	Effective risk (ER_B): .004
6. Breach detection	Audit log analysis; automated alarms; statistics utility results analysis	Effectiveness (CE_{AD}): .9
7. Breach recovery	Turning off firewall services; firewall shutdown	Effectiveness (CE_{BR}): .9
8. Harm	Loss of resources, time, and money.	Potential loss (PL_H): \$9,100 Total effective risk (ER_T): .001 Total effective loss (EL_T): \$6.57

Table 6 addresses a very different type of scenario: human error on the part of the firewall administrator. Despite the best of intentions on the administrator's part, he or she will make approximately three misconfigurations per year that the firewall will not prevent, of which a hacker could take advantage. Note that the total effective risk is 45 times higher than in the previous scenario, and the total effective loss per year is 27 times higher.

The two examples were chosen for their illustrative capabilities. Summing the Total Effective Loss values for all eight-stage event chains results in the average dollar amount lost per year due to the attack scenarios analyzed. After approximately 80 tables, each addressing an attack scenario, it becomes clear where weaknesses exist, and where additional security measures are needed.

Table 6. Human Error Scenario: Administration of ftp Access Controls

Stage	Instance	Effectiveness, likelihood, or potential loss level
1. Attack obstruction	Security education: system administrators are educated in the importance of the security policy and the procedures to adhere to it.	Effectiveness (CE_{AO}): .9
2. Attack scenario	System administrator inadvertently misconfigures ftp access controls.	Likelihood (PR_A): 3.00
3. Attack detection	User detection: system administrator realizes mistake, or co-worker notices misconfiguration.	Effectiveness (CE_{AD}): .4
4. Attack recovery	Firewall reconfiguration: system administrator corrects ftp access controls.	Effectiveness (CE_{AR}): .999
5. Security breach	Internet hacker discovers flaw, deletes files in ftp site.	Effective risk (ER_B): .18
6. Breach detection	Audit log analysis; user detection of file modification	Effectiveness (CE_{AD}): .75
7. Breach recovery	Firewall reconfiguration: system administrator resets access controls and restores ftp files.	Effectiveness (CE_{BR}): .999
8. Harm	Loss of ftp site resources and time to restore.	Potential loss (PL_H): \$4,000 Total effective risk (ER_T): .045 Total effective loss (EL_T): \$181

3. LESSONS LEARNED

"Firewalls are the wrong approach. They don't solve the general problem, and they make it very difficult or impossible to do many things. On the other hand, if I were in charge of a corporate network, I'd never consider hooking into the Internet without one. And if I were looking for a likely financially successful security product to invest in, I'd pick firewalls."

- Charlie Kaufman [6]

We couldn't agree more. Even when the firewall is supplemented with procedural defenses which rely on the users and administrators, the effective risk is still non-zero. At the end of each of our assessments, our customers all learned this lesson. The following are the additional lessons we learned.

3.1 A False Sense of Security

Firewalls give people the feeling that their systems on the internal network are secure, which leads to a sense of complacency. People feel they can "relax." Instead, the firewall has allowed access between the internal and external networks that users would normally feel a little less comfortable about. Ironically, internal network users should be even more concerned. The comfort provided by the firewall will tend to increase the flow of message traffic. The result is that all of the standard security precautions, e.g. running virus checkers on files that have been brought across the network, and being leery of e-mail that has been received from unknown sources, must be done with more consistency. The primary function of a firewall is to provide a buffer from external attack. Until firewall-to-firewall

authentication mechanisms are in place, we still suffer the consequences of inside users having access to sensitive information and the ability to send that information externally. The opportunity for error is high and there's no way to prevent that from happening other than through training and awareness classes. Gasser highlighted this issue in 1988 [11] by stating "Fads in the computer security area can have a serious negative effect on the overall progress to achieving good security because progress stops when people think they have the answer." Firewalls are inherently a crutch. By giving us a sense of protection from the external network, they allow us to put off addressing long-standing security issues with the systems on the internal network, such as lack of comprehensive access control enforced at the enterprise level or sensitivity checks on outgoing information.

RSA Data Security, Inc. is negotiating with leading firewall and TCP/IP stack vendors to create a security standard that could eliminate a major barrier to building virtual private networks (VPNs) on the Internet [12]. Even with this in place, it means you are still extending your trust to network and assuming it is trustworthy. For example, if a hacker has penetrated the other firewall's internal network, e.g. through a modem, and is communicating out through the other firewall, this new level of trust actually poses a threat to all systems which communicate with the firewall.

3.2 The Relationship between the Security Policy and the Firewall

Assessing this example firewall highlights the requirement that a security policy must be in place before the methodology can be applied. This requirement gives rise to two problems. The first is that many organizations, particularly commercial businesses suddenly coming to grips with the risks of being attached to the Internet for the first time, are in imminent danger. The immediacy of their need for security overrides the rational requirement for a well-reasoned, comprehensive policy. It's not even clear that many of the responsible policy-makers would know how to state their policy requirements. The second problem comes in translating between the security policy and the firewall implementation. Since the policy maker and the firewall administrator are usually different individuals, they may be unclear on the precise impact of their own decisions on each other's domains. In addition, administering a firewall requires making frequent, small changes to the configuration, effectively changing the firewall's security policy dynamically. Fortunately, a firewall, unlike many other security mechanisms, is well encapsulated. This leads us to an interesting proposal to firewall makers.

5. FUTURE DIRECTIONS

We recommend automating configuration of the firewall in conjunction with specification of the security policy. We envision a tool which presents the policy maker with potential policy statements. Statement selection would produce two outputs: 1) a human-readable description of the firewall security policy for the policy maker and the end users; 2) the associated configuration for the firewall. The tool

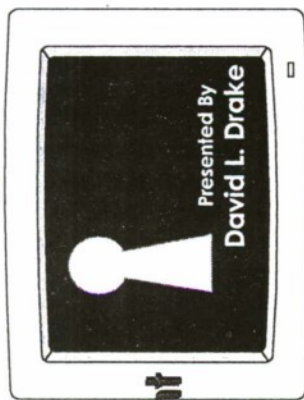
should also provide conflict resolution. Either selection of conflicting policy statements should be automatically prevented or they should be flagged as errors for the policy maker to resolve manually. In addition, direct changes to the configuration by the system administrator would be prohibited. Changes would be made through the same tool which would notify the administrator if an attempted change would violate the prescribed security policy.

Notice that we have come full circle to the fact that the biggest risk to a secure environment is still people. The proposed tool removes some of the potential for human error in the administration of a firewall.

REFERENCES

- [1] "The Security-Specific Eight Stage Risk Assessment Methodology," David L. Drake and Katherine L. Morse, *Proceedings of the 17th National Computer Security Conference*, 1994. Updated and republished in *Datapro Reports on Computer Security*, McGraw-Hill, 1995.
- [2] DoD 5200.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985.
- [3] FIPS PUB 31, *Guidelines for Automatic Data Processing Physical Security and Risk Management*, U.S. Department of Commerce, National Bureau of Standards, June 1974.
- [4] FIPS PUB 65, *Guidelines for Automatic Data Processing Risk Analysis*, U.S. Department of Commerce, National Bureau of Standards, 1 August 1979.
- [5] *Firewalls and Internet Security*, William R. Cheswick and Steven M. Bellovin, Addison-Wesley, 1994.
- [6] *Network Security: Private Communication in a Public World*, Charlie Kaufman, Radia Perlman, and Mike Speciner, Prentice Hall PTR, 1995.
- [7] *Computer Communications Security: Principles, Protocols and Techniques*, Warwick Ford, Prentice Hall PTR, 1994.
- [8] "Addressing Threats in World Wide Web Technology," Kraig Meyer, Stuart Schaeffer, Dixie Baker, IEEE Symposium on Computer Security, 1995.
- [9] "Security and the World Wide Web," David I. Dalva, Data Security Letter, Trusted Information Systems, June 1994. Available on the WWW:
<http://www.tis.com/Home/NetworkSecurity/WWW/Article.html>
- [10] *E-Mail Security: How to Keep Your Electronic Messages Private*, Bruce Schneier, John Wiley & Sons, Inc., 1995.
- [11] *Building a Secure Computer System*, Morrie Gasser, Van Nostrand Reinhold, 1988, pg. 12.
- [12] "Group Seeks Firewall Security Standard," Nick Wingfield, *InfoWorld*, Vol. 17, Issue 41, October 9, 1995.

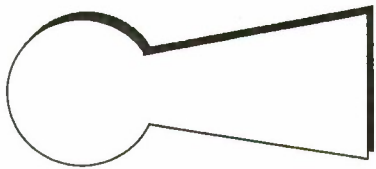
Applying the Eight-Stage Risk Assessment Methodology to Firewalls



NATIONAL INFORMATION SYSTEMS SECURITY CONFERENCE • OCTOBER • 1996

Outline

- Objectives
- The Eight-Stage Methodology
- Example Analysis of an Eight-Stage Event Chain
- Lessons Learned
- Future Directions



Objectives

- Demonstrate Eight-State Methodology's applicability to firewall assessments
- Highlight flaws in existing security assessment methodologies used for firewalls
 - Treating firewalls a special case assessment
 - Evaluating all countermeasures against all threat/vulnerability pairs
- Identify systemic firewall issues

The Eight-Stage Methodology



- | Assault | System Defense |
|---|---|
| <ul style="list-style-type: none"> • Security Incursion <ul style="list-style-type: none"> – Attack – Security Breach – Resultant Harm | <ul style="list-style-type: none"> • Prevention <ul style="list-style-type: none"> – Attack Obstruction • Detection <ul style="list-style-type: none"> – Attack Detection – Breach Detection • Recovery <ul style="list-style-type: none"> – Attack Recovery – Breach Recovery |



Example Analysis of Eight-Stage Event Chain

Stage	Instance	Effectiveness, Likelihood or Potential Loss Level
1. Attack Observation	Services written with secure feature; Firewall's use of secure version of sendmail.	Effectiveness (CEAD): .96
2. Attack Scenario	Hacker attempts to use the sendmail security holes to gain access to firewall.	Likelihood (PR _A): 2.0
3. Attack Detection	Audit log analysis; automated alarms.	Effectiveness (CEAD): .9
4. Attack Recovery	Turning off firewall services; firewall shutdown.	Effectiveness (CEAD): .9
5. Security Breach	Hacker gains access to firewall CPU time, system files and disk space.	Effective Risk (ER _B): .004
6. Breach Detection	Audit log analysis; automated alarms; statistics utility results analysis.	Effectiveness (CEAD): .9
7. Breach Recovery	Turning off firewall services; firewall shutdown.	Effectiveness (CEBR): .9
8. Harm	Loss of resources, time and money	Potential Loss (PL _H): \$ 9,100 Total Effective Risk (ER _T): .001 Total Effective Loss (EL _T): \$ 6.37



Lessons Learned

- Assessments were able to verify & quantify intuitive concerns
 - Firewall administration is too frequent & error prone
 - Firewalls give users & administrators a false sense of security
 - Firewalls usually require other hardware & software components to be inside the TCB
- Security policy implicitly changes every time firewall is administered



Future Directions

- Automate security policy enforcement
 - Input security requirements to firewall
 - Report and/or resolve conflicts
- Automate security policy documentation
 - Output security policy document
 - Output configuration for administrators
 - Output user procedures

LESSONS LEARNED:
AN EXAMINATION OF CRYPTOGRAPHIC SECURITY SERVICES
IN A FEDERAL AUTOMATED INFORMATION SYSTEM

Jim Foti
Donna Dodson
Sharon Keller
NIST, Building 820, Room 414, Gaithersburg, MD 20899

1. Introduction

Working with other agencies, the National Institute of Standards and Technology (NIST) recently completed a review of the security services implemented in an automated information system. During the review, several implementation flaws of the cryptographic security services were discovered which created vulnerabilities in an otherwise robust system. Based on findings during the review, recommendations were described and implemented to correct the security flaws and enhance the information system security already implemented in the system. The concerns described in the review could easily occur in other applications. Likewise, the recommendations provided by this review could be used by others to address security issues in other automated applications and systems.

1.1 Description of the System

To process information more efficiently, an agency recently developed an automated information system to replace and update its paper-based processing of work requests and approvals, in addition to the accounting associated with those requests. The system is based on a large relational database where electronic forms and user-provided data are stored in centrally located UNIX mainframes. A Wide Area Network is used to transmit information between users, and from the mainframes to PCs, so that users can manipulate and view the data and perform cryptographic security functions. There are currently 5000 system users, and it is projected that there will be 40,000 users by the end of 1997. Although the majority of the users are located in the United States, there are several sites in other parts of the world.

1.2 Use of Cryptography

Like many administrative applications, a replacement for handwritten signatures was required to totally automate this system. The agency also identified requirements for authentication and confidentiality; cryptography was employed to provide these security services. Because public key cryptographic standards were not available to the government during the design of the system, integrity, authentication, confidentiality, electronic signatures, and key management services were based on secret-key cryptography.

Key management is an essential component of the system, because it provides the foundation necessary to securely generate, store, distribute and translate keys. One of the fundamental principles for protecting keys is the practice of split knowledge and dual control. As defined in

ANSI X9.17-1985, *Financial Institution Key Management (Wholesale)* [1], split knowledge is "a condition under which two or more parties separately have key components which, individually, convey no knowledge of the resultant cryptographic key. The resultant key exists only within secure equipment". Dual control is explained in the standard as "a process of utilizing two or more separate entities (usually persons), operating in concert, to protect sensitive functions or information." Split knowledge and dual control were implemented in the system to protect the central storage of user keys, secure the distribution of user tokens, and initialize all cryptomodules in the system to "authorize" their use in performing cryptographic functions within the system.

Central sites also play an important role in key management. ANSI X9.17 relies on Key Management Facilities and Key Translation Centers to manage secret keys and translate those secret keys for decryption and signature verification. In public-key systems, central sites typically include a Certification Authority (CA), which is an entity that issues and revokes public key certificates, and may even generate key pairs. In either case, whether in a secret- or public-key system, the security of the central sites is critical to the overall cryptographic security of the system.

2. Findings and Recommendations

The recommendations listed below address the successful addition of cryptographic security to an automated information system. They are based on the review of the system described in section 1, as well as experience gained by NIST in other cryptography-related activities. These recommendations could be applied to many systems implementing cryptographic security services, whether the type of cryptography being used is secret- or public-key based.

2.1 General System Recommendations

- *Have cryptographic modules tested before distributing them throughout the system.*

Cryptographic services are provided using cryptographic modules (cryptomodules), which may include capabilities such as signature generation and verification (possibly involving key notarization), encryption and decryption, key generation, key distribution, etc. Examples of cryptomodules are smartcards, PCMCIA cards, PC adapters, and software modules, among other possible hardware, software, or firmware implementations.

If a large number of cryptomodules are needed to provide security services in a system, then an undetected error in a cryptomodule's design could potentially affect the performance of a cryptographic function for every user in the system. For example, key notarization (for secret keys) might be done improperly by a cryptomodule, or verifications of a chain of public key certificates might not function correctly. Key notarization helps ensure that no party other than the signer of the data can use the data key to sign or encrypt information. Likewise, verifying a chain of public key certificates helps a signature verifier determine if a signature was generated with a particular key. If either of these functions were to be implemented incorrectly in a cryptomodule, the

potential for the dissemination of weak cryptography could be introduced into the system, possibly allowing for signature forgery or the verification of invalid signatures.

This shows the importance of testing a cryptomodule before using it to provide cryptographic security services in a large system. Currently, Federal agencies are required to procure cryptomodules which have either been validated under the Cryptographic Module Validation (CMV) Program or submitted to an accredited laboratory for CMV testing. A series of tests are run on cryptomodules to test for conformance to FIPS PUB 140-1, *Security Requirements for Cryptographic Modules* [2]. These tests encompass features such as physical and operating system security, roles and services, and others. Under the CMV testing, cryptographic algorithms are tested for conformance to standards such as the Data Encryption Standard [3], Digital Signature Standard [4], and Secure Hash Standard [5]. The algorithms are exercised to detect implementation flaws, by performing tests which compare results generated by the implementation against known values and values generated by a reference implementation. Such testing would help detect implementation flaws in a cryptomodule's design.

- *Make use of cryptographic services as much as possible.*

By consistently replacing traditional methods of secure operation with cryptographic methods, the security and efficiency of a system improves dramatically. Benefits from implementing electronic or digital signatures include reducing the possibility of forgery, reducing processing time, and decreasing the burden of maintaining "traditional" paperwork. A system implementing cryptography will naturally generate new documentation, and the cryptographic technology should be applied in handling that documentation. Security officers, for example, may have to generate and sign requests for keys or cryptographic modules. Instead of using paper forms, electronic forms could be generated, signed, and sent to the appropriate parties, who can verify the signatures and act on the request in a very timely manner.

- *Provide consistent documentation and training to all system users, and place emphasis on educating them about cryptography.*

It is particularly important that all users understand the system they are using, and they should be aware of their responsibilities and the procedures they must follow in ordinary as well as unusual circumstances. These procedures should be standard among all sites in the system. Of special importance are the central sites, where security officers are responsible for equipment that might generate and manage keys for system users. If no standard set of procedures is followed, weaknesses may be introduced into the system.

2.2 General Key Management

- *Cryptographic keys may need special physical protection.*

If keys or key components are stored on a token (e.g., floppy disk, PCMCIA, smartcard, etc.), this token may have to be stored in a special manner to prevent unauthorized individuals from accessing the key or key component. For example, if key components for starting a Certification Authority or Key Management Facility are stored on tokens which are secured in a safe, multiple people might have access to this token. Therefore, additional protection is needed for each token, possibly by using a tamper-evident envelope, to enable the token's owner to determine if a token was used by another person.

- *Make sure that users are aware of their liabilities and responsibilities, and that they understand the importance of keeping their keys secure.*

The security of cryptographic keys in an electronic or digital signature system is the foundation of a secure system, therefore users *must* maintain control of their keys! Users must be provided with a list of responsibilities and liabilities, and each user should sign a statement acknowledging these concerns before receiving a key (if it is a long-term, user-controlled key). If different user types (e.g., security officer, regular user) are implemented in a system, they should be aware of their unique responsibilities, especially regarding the significance of a key compromise or loss.

- *Timeout features are important for protecting keys from compromise or misuse.*

A timeout feature for a cryptographic module or token is important, to minimize the possibility of an unauthorized individual accessing an "active" cryptomodule and using its cryptographic keys. This could happen if a cryptomodule is left unattended by a user who has authenticated to it and loaded his cryptographic keys. One alternative is to force a user to periodically re-authenticate herself to a cryptomodule, rather than allow her to stay logged in for an indefinite amount of time. For sensitive applications, it may be necessary to restrict the hours during which they can take place.

2.3 Key Management Facility / Certification Authority

- *Maintaining control of central or root keys from the time of generation is critical.*

Central or root keys are most likely to be used in sensitive applications such as encrypting user keys, signing a central key database for integrity, binding a key pair to a user, or generating user keys. If these keys are compromised, a complete system compromise becomes a very real threat. It is essential to maintain the security of these central keys from the very beginning - the generation process. No one but the proper owner(s) of a key or key component should ever be able to use that key or key component. If split knowledge and dual control are a requirement for central or root keys, then a failure to maintain split knowledge and dual control of those keys at *any* time in their lifecycle could present both a security problem and a potential system compromise.

- *Keep a log of when root keys are used.*

A record should be maintained of every instance that a central/root key is used. This should be an automated feature that is built into the system.

- *Sign all centrally stored data and encrypt sensitive data, such as secret and private keys.*

All centrally stored data that is related to user keys should be signed for integrity, and possibly encrypted for confidentiality (all secret and private keys should be encrypted). Individual key records in a database - as well as the entire database - should be signed. To enable tamper detection, each individual key record should be signed, so that its integrity can be checked before allowing that key to be used in a cryptographic function. When signing the entire database, at least the important fields that do not change regularly should be signed (this allows for faster verification).

- *Prepare for the possibility of compromise!*

It is imperative to have a contingency plan for the compromise or suspected compromise of central/root keys or key components at a central site; this should be established before the system goes "live". The contingency plan should address what actions should be taken with system software and hardware, central/root keys, user keys, previously-generated signatures, encrypted data, etc.

- *Sign and verify the code that implements the cryptographic functions.*

Software at the central key management site should be electronically signed and periodically verified to check the integrity of the code. This provides a means of detecting the unauthorized modification of system software. Within a cryptomodule, this feature of generating and verifying a cryptographic checksum is required by FIPS PUB 140-1.

- *A system implemented for a Government agency should have its centrally stored keys and system software controlled by Government employees.*

Proper control of central/root keys and key management software and hardware is critical to the security of the system. Federal employees should be in control of this material for a system operated for the Federal government. Once the system goes live, *unlimited* access to central data, code, and cryptomodules should *not* be given to non-government employees, including those who were contracted to develop and/or maintain the system. It is understood, though, that the agency may need outside assistance in maintaining the system.

- *Use different types of central and root keys, where possible, to maximize the scalability of the system and the integrity of cryptographic data.*

Different "types" of root keys might be implemented to 1) bring up a new system, 2) initialize a new central site, or 3) serve as backup keys for the same central site. It is very important to have backup copies of central/root keys, since the compromise or loss

of those components could prevent access to keys in the central database, and possibly deny system users the ability to decrypt data or perform signature verifications.

- *Be aware of security issues when migrating from a prototype to a live system.*

When moving the system from a prototype to a live phase, the safest strategy is to generate new central/root keys and reissue keys for other system users. However, if it is not feasible to do this, then prior to migration a review of the generation, distribution, and storage procedures used for the root keys should be performed, to ensure that their security was maintained throughout their lifecycle. Otherwise, a security flaw or compromise in the prototype phase could be passed on to the live system.

- *Keep the KMF/CA flexible for scalability*

Allow for the possibility of multiple "central" sites. More than the original number may be required if more users are added to the system. Ramifications on the root keys should be considered, including 1) how are they stored, 2) how are root keys to be generated for and distributed to the new central site, and 3) how will database information be communicated to the new central site and used by holders of the new root keys.

2.4 Key Distribution

- *If a key is stored on a token, and a PIN is used to access the token, then only that token's owner should ever have possession of both the token and its corresponding PIN.*

This applies to root security officers who may generate a token and its PIN, as well as any intermediaries. To prevent a courier from having sole control of both items, security officers should distribute the token and PIN in separate mailings (in separate packages mailed on different days). Receipt of each item should always be confirmed to the original sender. A failure to maintain control of this token and PIN could lead to a key compromise and the misuse of cryptographic functions within the system.

2.5 Key Storage and Destruction

- *Determine reasonable lifetimes for keys associated with different types of users.*

Users with different roles in the system should have keys with lifetimes that take into account the users' roles and responsibilities, the applications for which the keys are used, and the security services which are provided by the keys (user/data authentication, confidentiality, data integrity, etc.). Reissuing keys should not be done so often that it becomes burdensome, however it should be performed often enough to minimize the chance of key compromise.

- *Archive user keys for a sufficiently long cryptoperiod.*

A cryptoperiod is the time during which a key can be used for signature verification or decryption; it should extend well beyond the lifetime of a key (where the lifetime is the time during which a key can be used to generate a signature and/or perform encryption). Keys should be archived for a lengthy cryptoperiod (on the order of decades, perhaps), so that they can be used to verify signatures and decrypt ciphertext at any point during that time.

- *Handle the deactivation/revocation of keys so that data signed prior to a compromise date (or date of loss) can be verified.*

It should be possible to designate a signing key as LOST or COMPROMISED, so signatures generated prior to a specified date can be verified. Otherwise, all data previously signed with a lost/compromised key would have to be reviewed and re-signed.

2.6 Signature Generation and Verification; Encryption and Decryption

- *Protect data prior to signature generation/verification and encryption/decryption. Be careful of how data is handled during these processes!*

Implementors should be very careful about how data is handled before it is signed/verified (encrypted/decrypted). If the data is stored on the computer where the cryptographic function is performed, this might not pose a problem. However, if data is stored in a central database and transferred to the computer only at the time the cryptographic function is to be performed, the data should be very carefully protected during transmission. If data is not carefully protected, then an intruder could potentially alter data before a signature is generated, without the signer's knowledge.

- *Before generating a signature, users should be able to view all data to be signed. It should be made obvious to users as to exactly what data a cryptographic function is applied to.*

User should be able to see all the data that is being signed, and it should be clearly marked for the signer. It is not always intuitive for a user to discern which data is included in a signature. Knowing what is encrypted is important, too - a user may be concerned if he knows that certain data is not being encrypted. It is not essential that all data being signed/encrypted should appear on one screen, but the user should at least be able to view all of the data before performing the cryptographic function.

- *Plan for the need of a user to re-sign data, in a tightly-controlled manner that is logged.*

Signature verification may fail due to a change in an organizational code, a form number, a person's last name, etc. These values might be more likely to change between signature generation and verification if they are pulled from a database to reconstruct a message. Strict controls should be put in place to restrict the use of the re-signing capability to specific situations and/or specific individuals (e.g., the original signer or a

database administrator acting on the original signer's behalf). The re-signing tool should allow a person to 1) examine what changed in the message content from the time of the original signature, and 2) decide whether or not the change warrants the generation of a new signature. All use of the re-signing tool should be carefully controlled and audited. Such an audit trail should minimally include: suspected cause of verification failure, whether or not the data was re-signed, who determined the data should be re-signed, who performed the re-signing, and the date/time of re-signing.

- *Determine what data fields must be protected using a cryptographic function.*

The implementor should be aware of what fields are being signed and encrypted. It may not be necessary for all fields in a form to be signed and/or encrypted. Limiting the data input to a cryptographic function may have a significant impact on the speed with which that function can be performed. Fields containing sensitive data should be identified, and then a determination should be made of what cryptographic functions should be applied to those fields: integrity, authenticity, and/or confidentiality.

3. Summary

The recommendations mentioned in section 2 of this document should be taken into consideration when cryptographic services for a system are being designed. However, they do not form a comprehensive list of issues that must be addressed. It is important to remember that adding cryptography to a system will not necessarily provide adequate security. Cryptography must be designed as an integrated part of the system, rather than as an add-on feature. The agency whose system was reviewed by NIST took the approach of designing cryptography into the system from the very beginning. For those situations where this cannot be done, cryptographic functions should be carefully added so that the security that they are intended to provide is not compromised.

4. References

- [1] ANSI X9.17-1985, *Financial Institution Key Management (Wholesale)*, American Banker's Association, Approved April 4, 1985, Reaffirmed 1991.
- [2] FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*, US DOC/NIST, January 11, 1994.
- [3] FIPS PUB 46-2, *Data Encryption Standard (DES)*, US DOC/NIST, Reaffirmed December 30, 1993.
- [4] FIPS PUB 186, *Digital Signature Standard (DSS)*, US DOC/NIST, May 19, 1994.
- [5] FIPS PUB 180-1, *Secure Hash Standard (SHS)*, US DOC/NIST, April 17, 1995.

INTELLECTUAL PROPERTY RIGHTS AND COMPUTER SOFTWARE

Dawn E. Bowman
DawnSheree@AOL.COM
©1996

ABSTRACT/ EXECUTIVE SUMMARY

The three goals of computer security, namely secrecy, integrity, and availability, are most commonly achieved through physical and electronic measures. However, laws also exist whose intent is to protect the secrecy, integrity, and availability of computer hardware, software, and data. The laws referred to here which govern computer security fall under the domain of intellectual property. Unfortunately, current domestic laws governing intellectual property as they relate specifically to computer security and computer software lag far behind the technology these laws are intended to protect. Most notably glaring is the fact that domestic laws are more timely than many of the foreign counterparts. Advances have been and continue to be made in the creation of required legislation and in the prosecution of criminal offenders, but the battle is just beginning. In order for any progress to be made in this area of the law, the justice system will be obliged to change from a reactive mindset to a proactive mindset. When and if this transformation can occur remains to be seen.

INTRODUCTION AND OVERVIEW

As previously mentioned, the main goals of computer security are to ensure secrecy, integrity, and availability of hardware, software, and data. The focus of this analysis, however, will consider the effect of the elements of security afforded under the current system of intellectual property laws as the laws relate specifically to software.

The objectives of this paper are as follows:

- To discuss the various types of intellectual property including patents, trademarks, and copyrights,
- To discuss the legal hurdles surrounding computer software and its place in the intellectual property arena,
- To provide a brief legislative history of the various areas of intellectual property,
- To ascertain the impact the current intellectual property laws is having on the economic condition of the U.S.,
- To compare and contrast the intellectual property laws in foreign countries with the U.S. counterparts,
- To consider the impact of current intellectual property laws as they relate to computer security, and
- To analyze trends for the future in the area of intellectual property as they relate to computer software.

INTELLECTUAL PROPERTY - WHAT IS IT?

The U.S. "intellectual property system" contains elements of both Federal and State law. Laws related to copyright, patent, and trademark all fall under Federal jurisdiction while laws concerning trade secrets are covered under State jurisdictions. Computer software law is distinguished from most other intellectual creations protected by intellectual property law in that different aspects of the software is eligible for protection by patent, copyright and trade secret laws. Each type of protection has advantages and disadvantages under the current laws.¹

Patent Law

A patent is a grant of a property right by the Government to the inventor "to exclude others from making, using or selling the invention."² A patent protects the device or process for carrying out an idea, not the idea itself.³ Within the category of patents, there are three types of patents available: (1) Utility patent, (2) Design patent, and (3) Plant patent.

A Utility Patent is limited to a process, machine article of manufacture, or a composition of matter that meets the criteria of being (1) novel, (2) non-obvious, and (3) useful. The rights associated with a patent prevent others from making, using or selling the patented invention or component thereof. A patent also protects against independent creation so that the holder of the patent does not need to prove that their idea was copied by another inventor; the fact that the invention has been patented is proof enough. A U.S. utility patent typically allows for 17 years of protection, although there may be exceptions to

¹U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (Washington, DC: U.S. Government Printing Office, May 1992).

²U.S. Department of Commerce, Patent and Trademark Office, *Basic Facts about Patents*.

³Pfleeger, Charles P. *Security in Computing*. New Jersey: Prentice Hall PTR, 1989.

this term. Legislation enacted last year scheduled for 1996 implementation sought to change the patent term from 17 years from date of grant to 20 years from date of filing. This legislation brought so much outcry from independent inventors that two bills were introduced last year, HR 359 by Rep. Dana Rohrabacher (R-Calif.) and S 284 by Sen. Robert Dole (R-Kan.) which would set the patent term at 17 years from grant or 20 years from filing, whichever is greater. These bills would supposedly ensure maximum patent protection for inventors whose applications languish unduly in the Patent Office before being issued as patents.

A Design Patent is available for surface ornamentation, configuration, or a combination of the two. Patent protection for designs is granted for a period of 14 years.

Finally, a Plant Patent is granted to any person who has invented or discovered and asexually reproduced any distinct and new variety of plant, including cultivated spores, mutants, hybrids, and newly found seedlings other than tuber-propagated plant or a plant found in an uncultivated state.⁴

Copyright Law

Copyright law in the U.S. protects the right of an author to control the reproduction, adaptation, public distribution, public works display, and public performance of original works of authorship of every kind, ranging from books to sound recordings.⁵ Copyright law protects the expression of an idea, not the underlying idea itself. Current copyright law provides for copyright protection for unpublished as well as published works. This is important for computer software, because it facilitates simultaneous use of copyright and trade secret protections. The published version of the copyrighted program can be distributed as "object code" whereas the "source code" may remain unpublished so as to protect the program's logic. Copyright does not extend to any procedure, process, system, or method of operation, regardless of its form. Rather, copyright is said to protect the expression in the program---which may include such program elements as source code, object code screen displays, etc. Unlike patents, copyright does not protect against independent creation.⁶ A fundamental goal of U.S. copyright law is to promote the public interest and knowledge. Although copyright is a property interest, its primary purpose was not conceived of as the collection of royalties or the protection of property; rather, copyright was developed primarily for the promotion of intellectual pursuits and public knowledge. Therefore the congressionally mandated grant of a limited monopoly for authors is based upon dual interests: the belief that the public should benefit from the creativity of authors and the belief that a copyright monopoly is necessary to stimulate the greatest creativity of authors.

Copyright law is a balancing act between intellectual promotion and property rights. The concept of copyright appears to be a paradox when considered in the context of freedom of speech given by the first amendment. It may seem that copyright expression restricts the freedom of information, however others may argue that to the degree that copyright protection stimulates, the restrictions are worthwhile. Much of the balance achieved between these two underlying principles is that of the fair use doctrine. The doctrine was codified in 1976 to become a part of the Copyright Act.

A subset of the copyright laws is the **Semiconductor Chip Protection Act of 1984 (SCPA)**. This act extends legal protection to a new form of subject matter, namely semiconductor chip mask works. Semiconductor chips may be defined as integrated circuits containing transistors, resistors, capacitors and their interconnection, fabricated into a very small single piece of semiconductor material. A mask work is a set of images fixed or encoded at a later stage of manufacturing, that produces the circuitry of the final chip product.⁷ According to legislative history, the Semiconductor Chip Protection Act was intended to combat the problem of chip piracy, as Congress perceived that the existing law failed to address that problem. The Chip Act is a *sui generis* law (a law of its own kind or class), creating a statutory scheme to provide property protection for chip products separate from and independent of the Copyright Act.

Trade Secret Law

Trade secret law protects the information in a patent until the patent is actually granted. In addition, information that is beyond what is required for inclusion in the patent to meet the "enablement" and "best mode" requirements can also be reserved for trade secret protection. Trade secret law also protects confidential business information against unauthorized use or disclosure and is based on statutory and common law and contractual provisions. A classic example of a trade secret is the formula for the popular soda, Coca-Cola. Similar to patents, trade secret law can protect that underlying idea of an invention, rather than any particular expression of that idea.⁸ Trade Secret law is one of the most widely used forms of legal protection for intellectual property interests in computer software. Numerous courts of a variety of U.S. jurisdictions have ruled that trade secret properly protects computer software. When software is distributed to relatively few customers, licenses establishing the confidential relationship and obligations necessary for maintaining the trade secret can be obtained through signed written

⁴U.S. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (Washington, DC: U.S. Government Printing Office, May 1992).

⁵*Finding a Balance* 56.

⁶*Finding a Balance* 60.

⁷*Finding a Balance* 75.

⁸*Finding a Balance* 13.

agreements. Developers of computer software have attempted to address the more difficult problem of maintaining trade secrecy in mass marketed software, extensive distribution of which might otherwise destroy requisite secrecy by what is known as a "shrink wrap" license. The shrink wrap license signals further secrecy, and is established by marketing software in a sealed package with a notice and a license agreement that is visible on the exterior of the package. The agreement generally provides that the user, by opening the package, is deemed to have accepted the license terms and conditions. The terms of such a license generally prohibit decompilation, disassembly or copying of a program for any reason except for use and backup purposes.⁹

One of the ways that trade secret law can be rendered ineffective is through the process of reverse engineering. This is a process whereby the finished object is studied to determine how the object is originally put together. Of course, the most obvious way to circumvent the protection of a trade secret is to expose the secret.

In summary, the U.S. intellectual property system is composed of several types of intellectual property, namely patents, copyrights, trade secrets, and trademarks. Trademark law does not particularly apply to protection of any aspect of computer software and is therefore beyond the scope of this discussion. The next section will address in more detail the various aspects of computer software in which each of these types of intellectual property has jurisdiction.

INTELLECTUAL PROPERTY AND COMPUTER SOFTWARE

There are intellectual property issues associated with four elements of a software program:

1. Program function - whether the algorithm is performed by the hardware or the software,
2. External design - the conventions for communication between the program and the user or other programs,
3. User interfaces - the interactions between the program and the user,
4. Program code - the implementation of the function and external design of the program.

Whether and to what extent software-related inventions are the subject of utility patent protection had been an issue for consideration by the courts since the early 1960s. The U.S. Supreme Court has examined the issue of patentability of software on a number of occasions, in the cases of *Gottschalk v. Benson*, *Parker v. Flook*, and *Diamond v. Diehr* attempting to delineate the limits of patentable subject matter with respect to "mathematical algorithms."

The scope of copyright protection for computer programs depends in part on the interpretation of Section 102(b) of the Copyright Act. There are a number of existing views of the application of existing law to user interfaces. One interpretation of the law is that user interfaces are inherently functional and therefore not copyrightable subject matter. The other view is that user interfaces may be protected by copyright because they could be thought to fall under the compilations or audio-visual works. Another approach to protecting user interfaces through copyright law is to consider the user interface as part of the program itself.¹⁰

Databases are protected under copyright law as compilations. Under the copyright law, a compilation is defined as a work formed by the collection and assembling of pre-existing materials of data that are selected, coordinated, or arranged in such a way that the resulting work as a whole constitutes an original work of authorship (17 USC Section 101). In April 1991, the Supreme Court "dropped a bomb" when it held in *Feist Publications Inc. v. Rural Telephone Service Company, Inc.* that the white pages of a typical telephone directory were not copyrightable. The decision sent shockwaves throughout the computer industry because of the questions it raised about copyright protection for other fact-based compilations, such as computer databases. The Supreme Court effectively reaffirmed that copyright originality requires a minimum level of human creativity which some databases may not meet.¹¹

LEGISLATIVE HISTORY vs. ECONOMIC REALITY

Evolution of Patent Case Law

During the early 1960s, the Patent and Trademark Office (PTO) faced a large backlog of patent applications and an average pendency of 4 years. In 1965, the President's Commission on the Patent System was established to address these problems and suggest revisions to the Patent Act. The PTO denied the patentability of computer programs in 1964, characterizing them as "creations in the area of thought." In 1966, the PTO attempted to formulate standards in the patentability of software, however nothing came to fruition until the Supreme Court finally considered the issue of the patentability of computer software in the case of *Gottschalk v. Benson*.¹² This case involved a request to patent a process for converting decimal numbers into binary. The case was rejected however by the High Court because it seemed to patent an algorithm or idea. In 1978, the Supreme Court again addressed the question of software patentability in the case of *Parker v.*

⁹Finding a Balance 75.

¹⁰Finding a Balance 75.

¹¹Patent Trademark and Copyright Journal 18 Apr. 1991.

¹²Finding a Balance 48.

*Flook*¹³. Although the Supreme Court ruled the subject in this case as not patentable, the decision left open the possibility that "a process is not unpatentable simply because it contains a law of nature or mathematical algorithm." In 1981, the case of *Diamond v. Diehr* came to the Supreme Court. This case won a patent for a process that used computer software, a well-known algorithm, temperature sensors, and a computer to calculate the time it took to cure rubber seals¹⁴. The Court reversed the decision of the Patent Appeals Court on the basis that claims are not disqualified from patentability because of the use of a mathematical equation and programmed digital computer.

Evolution of Copyright Law

The Copyright Act was enacted in 1790, and has endured many revisions since then, with the most recent overhaul happening in 1976 with legislation that modified the term of copyright and codified the "fair-use" concept as a limitation on the exclusive rights of the holder. The fair use doctrine allows the unauthorized use of certain copyrighted material in comment and criticism, news reporting and classroom teaching. This doctrine allows the courts to bypass an inflexible application of copyright law when under certain circumstances it would impede the creative activity that the copyright law was supposed to stimulate. Congress has created statutory regulations of a list of factors that courts should consider in making their fair-use determination. The four factors set out in the statute are:

1. the purpose and character of the use, including whether such use is of a commercial nature or is for non-profit educational purposes;
2. the nature of the copyrighted work;
3. the amount and substantiality of the portion used in relation to the copyrighted work as a whole; and
4. the effect of the use on the potential market and value of the copyrighted work (17 USC 107 (1988)).¹⁵

In the late 1970s, Congress established the National Commission on New Technological Uses of Copyrighted Works (CONTU) to make recommendations for computer copyright legislation specifically related to computer software or programs, databases, and works created by the use of computers. In 1980, following recommendations made by CONTU, legislation explicitly extended copyright protection to computer programs. Therefore computer programs became copyrightable as "literary works" as defined in 17 USC 101. The term "computer program" is also defined in section 101 as "a set of statements or instructions used directly or indirectly in a computer in order to bring about a certain result."

Software Piracy

Gail Penner, who testified on behalf of the Software Publishers Association, said that in 1990 the U.S. software industry lost \$2.4 billion in revenues in the United States alone, and between \$10 and \$12 billion worldwide. "Civil remedies are not adequate to combat software piracy", she declared, "because the pirates can easily go underground for a while and pop up again in another guise. Moreover, because counterfeiters rarely keep business records, civil discovery methods to secure information about infringers' assets, for instance, do not work."

Creators of commercial software are concerned about their profitability. An important rationale for creation and enforcement of intellectual property rights is to give commercial software developers adequate market incentives to invest time and resources needed to produce and disseminate innovative products. Illegal copying of software results in financial losses to U.S. software firms both directly, through loss of sales and or royalties, and indirectly, through loss of investment opportunities.

Criminal Penalties for Copyright Infringement

In the early 1980s, according to Section 506(a) of the Copyright Act, willful infringement of the Copyright Act for commercial advantage or private financial gain was punishable according to the provisions of 18 USC 2319. In 1982, PL97-180 was enacted to toughen the criminal penalties under Title 18 for pirating and counterfeiting copyrighted sound recordings, motion pictures or other audiovisual works. This legislation was directed at commercial counterfeiting which had been only a misdemeanor rarely pursued by federal prosecutors. Section 2319 of Title 18 now provides maximum felony penalties of five years imprisonment of anyone who, within any 180 day period, illegally reproduces or distributes at least 1,000 copies of a copyrighted sound recording or at least 65 copies of a copyrighted motion picture or other audiovisual work. Where more than 100 and less than 1,000 copies of sound recordings are involved, or more than seven and less than 65 motion pictures or other audiovisual works are involved, the maximum prison time is two years and the fine is the same. For criminal infringement of other works, the maximum penalties are one year of imprisonment and/or \$25,000.¹⁶

¹³*Finding a Balance* 50.

¹⁴Pfleeger, Charles P. *Security in Computing*. New Jersey; Prentice Hall PTR, 1989.

¹⁵*Finding a Balance* 62.

¹⁶*Patent Trademark and Copyright Journal* 11 Jun. 1992.

Once the former Soviet Union disintegrated, global economic competition replaced military confrontations as the principal threat to national prosperity. This development coincided with the software industry's growing aggravation with widespread copying of computer programs. The industry saw the example the film and recording industry provided, where felony prosecutions had brought piracy under control, and asked Congress for the same protection. Hearings were held in the Senate first and the Senate Report candidly acknowledged that the bill was motivated by concerns for the global competitive position of the U.S. To put the global position of the U.S. in perspective, note that the world market for computer software is currently estimated at \$70 billion per year. U.S. companies now hold a 70 percent share of the world software market, generating about \$50 billion in sales.¹⁷

In April 1992, Senator Orrin Hatch (R-Utah) introduced a bill (Section 893) to add computer programs to the list of works that are the subject of criminal penalties under 18 USC 2319. Thus, under Senator Hatch's proposal, Section 2319(b)(1) of Title 18 would be amended to provide fines of up to \$250,000 and imprisonment of up to five years for willful infringement for commercial advantage or private financial gain "that involves the reproduction and distribution, during any 180 day period, of at least 50 copies infringing the copyright in any one or more computer program (including any tape, disk, or other medium embodying such programs)." For infringements involving more than 10 but less than 49 copies, an amended Section 2319(b)(2) would provide fines of up to \$25,000 and imprisonment of up to one year.¹⁸ These harsher punishments match those provided under similar legislation enacted in 1982 to deter counterfeiting of records, tapes and films.¹⁹

An amendment to the Senate Bill 893 was proposed in September 1992 by Subcommittee Chair Representative William J. Hughes (D-New Jersey). His substitute amendment proposed to make the felony/misdemeanor turn on the retail value of the copies, rather than on the number of copies made. The threshold retail value for a felony offense would be \$5,000. The felony penalty would be two years of prison, and repeat offenders would be subject to a maximum of five years imprisonment.²⁰

The Clinton Administration has also made attempts to address the shortcomings of current legislation as it relates to information technology. The Working Group on Intellectual Property Rights, the Clinton Administration's Information Infrastructure Task Force, released a report in July 1994 on Intellectual Property and the National Information Infrastructure (NII). The report included recommended changes to existing copyright law to accommodate advances in information technology.²¹ If these recommendations are approved by Congress there may be new restrictions on what one can and cannot do on the Information Superhighway.

Mr. Bruce A. Lehman, Assistant Secretary of Commerce as well as the Commissioner of the Patent and Trademark Office, heads the intellectual property working group that prepared the report as part of the Clinton Administration's NII Task Force, which is seeking to establish a blueprint for the Information Superhighway. Experts, such as Gail Penner of the Software Business Alliance, an 1100-member trade association, say that the copyright law has not kept pace with technology especially in the area of digitization. This technology allows for the creation of an innumerable number of copies of a particular work that are virtually identical to the original. Lehman's plan is to change the Copyright Act to include digital transmissions, making it clear that on-line distribution rights are held by the copyright owner the same way they are on paper. It would also become a crime to disable anti-copying technology.²²

Other recommendations contained in the report propose that the U.S. copyright law be amended to explicitly protect electronic information and its transmission. In addition, the term "publication" would be given a broader definition to include distribution by transmission and the first-sale doctrine (which allows the owner of a lawfully made copy to sell or dispose of that one copy) would be rewritten to exclude electronic transmission. Although some of the changes recommended in the report are clearly needed, a number of groups (libraries and educational groups) have expressed concerns in public hearings that there must be a balance between "fair use" and protecting the copyright holders needs. Otherwise, first amendment rights may be seriously jeopardized in cyberspace. Publishers, in general, have been supportive of the recommendations. Clearly, the proposed changes to the copyright law could have broad consequences.²³ Various industry groups have said there is a pressing need for such measures in light of the growing use of computers and computer networks that can easily copy and disseminate text, pictures, sound and video images that exist in digital form.

In September 1995, a bill was introduced in the House of Representatives which would amend Title 17 of the Copyright Act, to adapt the copyright law to the digital, networked environment of the National Information Infrastructure, and for other purposes. The short title of this Act may be cited as the "NII Copyright Protection Act of 1995."²⁴

Corporate executives are also showing their support for new information technology laws. Barbara A. Munder, Senior Vice President of the McGraw Hill Companies, Inc. testified on behalf of the Information Industry Association before the House Committee on the judiciary to provide support for the proposed House Bill HR 2441 NII Copyright Protection Act of 1995. Ms. Munder stated "We do not and cannot offer more because there is too great a risk to our valuable intellectual

¹⁷Spanner, Robert A. "The Brave New World of Criminal Software Infringement Prosecutions." *The Computer Lawyer* November 1995, Page 1.

¹⁸"Senate Passes Bill to Stiffen Criminal Penalties for Software Infringement." *Patent Trademark and Copyright Journal* 4 Jun. 1992: 121.

¹⁹"Legislation, Copyrights." *Patent Trademark and Copyright Journal* 9 Apr. 1992: 511.

²⁰"House Panel Amends Bill on Criminal Penalties for Software Infringement." *Patent Trademark and Copyright Journal* 17 Sept. 1992: 485.

²¹United States. Information Infrastructure Task Force, *National Information Infrastructure Report*, 172 D 128, 1995.

²²Kantrowitz, Barbara. "My Info is NOT Your Info" *Newsweek* July 1994: 54.

²³"Free Speech and Copyright in Cyberspace: Legal Issues Surrounding the Internet." *On-line Libraries and Microcomputers* Mar. 1995.

²⁴Full Text of Bills, 104th Congress, 1st session, H.R. 2441, September 29, 1995.

property in an environment where the culture and technology offer so little protection for the rights of content producers." "...without effective protection, we cannot risk our hard work and investment in cyberspace where it is so easy to copy, retransmit and alter our property without our permission, and often without our knowledge."²⁵

An example of where the current system has failed the public is apparent in the case of *United States v. LaMacchia*. In the 1994 case of *United States v. LaMacchia*, an MIT student set up a computer bulletin board on the Internet and invited users to upload copies of popular software which could then be downloaded for free use by other users. Although LaMacchia's scheme allegedly caused losses of over \$1 million to software copyright holders, he could not be sued for criminal copyright infringement because there was no evidence that he sought or derived any 'commercial advantage or private financial gain.' In response to this case, Sen. Patrick Leahy (D-Vt.) on August 4, 1995, introduced a bill (S 1122) to reinforce criminal copyright infringement provisions for infringement of works worth \$5,000 or more, even where the infringer neither sought nor derived any commercial advantage or financial gain. The bill which also expressly prohibits "assisting others" in the reproduction or distribution of an infringed work, would close a loophole in the law that became apparent in the 1994 case which involved software infringement on an Internet computer bulletin board.

The bill proposed by Sen. Leahy would also continue to encourage growth of the National Information Infrastructure by ensuring better protection of the creative works available on-line. The definition of "financial gain" would be changed to include the bartering for, and trading of pirated software. It would also amend Section 5-7(a) of Title 17 to extend the statute of limitations for criminal copyright infringement from three to five years, as is currently provided under Title 18 for sound recording and counterfeit tracking.

Criminal sanctions would attach under Section 506(a)(2) if only a single copy were made of a copyrighted works that meets the \$5,000 monetary threshold. For infringed works with a value between \$5,000 and \$10,000 the offense would be a misdemeanor; for works worth over \$10,000, the offense would be a felony. The penalty provisions at 18 USC 2319 would be amended to conform with the proposed amendments to Section 506(a)(2). The new offense under Section 506(a)(2) of willfully infringing works worth \$10,000 or more would be punishable by a fine and up to five years imprisonment; for infringement of works worth between \$5,000 and \$10,000, the punishment would be up to one year in prison and a fine. Repeat offenders would be punished by up to ten years imprisonment and a fine. A new Subsection 2319(e) would be added requiring that victims of criminal copyright infringement be given the opportunity to provide a victim impact statement to the probation officer preparing the pre-sentence report.²⁶

All of these events, such as the National Information Infrastructure Task Force, pending bills in the House and Senate for increased penalties of infringement, and expert testimony by those individuals in the information technology industry are all good starts in the process of reforming the current intellectual property laws, specifically as these laws relate to information technology and computer software. However, a good start does not guarantee a good finish. It has taken many years for the 'powers that be' to realize the importance of up to date, timely information technology laws. The key to success lies in keeping ahead of the technology which will require an even greater commitment.

INTELLECTUAL PROPERTY ABROAD

Patent Law in Foreign Countries

The patent, trademark and copyright laws that are currently in place in the U.S. may differ substantially from the corresponding laws in foreign countries. Most of the European, Asian, and Latin American countries with established patent law systems do not offer protection for software programs. Some of the countries expressly exclude software from patentability such as Brazil, France, and Switzerland. In contrast, other countries are silent and therefore leave open the possibility of software patentability such as Japan, Taiwan, Korea, and Thailand.

Japan and Taiwan have granted patents for certain computer programs, especially if the computer program is described in conjunction with a method or computer in which the program is used in the specification of an application.

The European Communities have agreed in their Software Directive that the prescribed protection of computer programs under copyright law does not prejudice the application of other forms of protection where appropriate. Computer software may be protected under patent law in addition to copyright in European Community members.²⁷

Copyright Law in Foreign Countries

The copyright laws in foreign countries are as varied as the nations themselves. A complete discussion of the nuances of the copyright laws of the major European, Asian, and Latin American countries is beyond the scope of this paper. However, Japan and Europe (as a whole) will be briefly discussed as the U.S. is involved in a great deal of trade with these countries.

²⁵"Prepared Testimony of Barbara A. Munder..." *Federal News Service* 7 Feb. 1996.

²⁶"Bill Would Strengthen Penalties for Criminal Infringement on Internet." *Patent, Trademark and Copyright Journal* 10 Aug. 1995: 368.

²⁷ *Finding a Balance*

In Japan, to ensure inclusion of computer programs as protectable subject matter of copyright, the Japanese revised copyright law to define computer programs as "a set of instructions for a computer which are combined in order to function (sic) the computer so that one result can be obtained." Under Japanese law, both source code and object code are copyrightable. Translation from source code to object code constitutes a reproduction of the source code. Japanese copyright law further provides that the author shall have the exclusive right to reproduce his work, as well as to translate, arrange, transform, dramatize, cinematize, or otherwise adapt his work. In principal, a person who possesses a copy of a program is prohibited from making another copy or adapting the original copy without the copyright owner's consent. However, like U.S. law, Japanese copyright law limits the scope of the author's exclusive right of reproduction regarding a program work, by allowing copies or adaptation to the extent deemed necessary for the purpose of using the work in a computer to be made by the owner of a program for his own use. The period of protection for computer software in Japan is life of the creator plus fifty years. For unpublished software, the copyright lasts fifty years after the creation of the work.

The European Communities have adopted a directive on the legal protection for computer software which must be implemented by each of the EC member states. This directive requires that software be protected by copyright as a literary work within the meaning of the Berne Convention.

Trade Secret Law in Foreign Countries

Japan is the only Pacific Rim nation whose law provides for trade secret protection. The Japanese law defines a trade secret as technological or business information useful for business activities, controlled as a secret, which is not publicly known art. Under the law, if a computer program properly qualifies as a trade secret, the owner of a computer program who is damaged or is likely to suffer damage by unauthorized use or disclosure of his program may require the offending party to stop the unauthorized use or disclosure of the program. The owner of a trade secret may request that the media on which the program is stored be destroyed. (However, since there are no "protective orders" in court proceedings, the secret may be lost as a result of bringing the litigation.) Unfair activity includes acquisition of a trade secret by stealing, deception, or threats, or acquisition from a third party while aware that the trade secret was originally acquired by an unfair activity.

The European Communities have agreed in their long debated Software Directive that the prescribed protection of computer programs under copyright does not prejudice the application of other forms of protection where appropriate. Thus, computer software is properly protected by trade secret in addition to copyright in European Communities member nations.

Foreign Treaties

In addition to the various analogous foreign patent, trade secret, and copyright laws discussed previously, there also exist several international treaties among various nations.

The Berne Convention for the Protection of Literary and Artistic Works is a multilateral, international copyright treaty. The purpose of the Berne convention is to bring nations together in an effort to protect, in as effective and uniform a manner as possible, the rights of authors in their literary and artistic works.

The General Agreement on Tariffs and Trade (GATT) is a multilateral trade agreement, entered into force in 1984, intended to promote freer trade among member countries. The GATT is the main instrument regulating trade among market economy nations of the world.

The World Intellectual Property Organization (WIPO) attempts to address and synchronize patent issues in foreign countries with those in the U.S. For example, legislation was recently introduced (February 1995) in both the House and the Senate to extend the term of copyright protection. House bill 989 and Senate bill 483 seek to extend the term of protection which is currently determined as the author's life plus fifty years. The bills seek to extend this period to the author's life plus seventy years. This increase in the term of copyright protection is designed to match the new European Union standard. Significantly, the increase in the copyright term protection would be applied retroactively, adding twenty years to the renewal terms of copyrights in their first term as of January 1, 1978.²⁸

Despite the international treaties currently in place, other laws exist which attempt to address those international situations where fairness may not be the name of the game. The 1988 Omnibus Trade and Competitiveness Act amended Section 182 of the Trade Act of 1974, to require that the U.S. Trade Representative annually identify countries that deny "adequate and effective" intellectual property protection and "fair and equitable" market access to U.S. businesses that depend on intellectual property protection. Countries deemed to have the most unacceptable standards are designated "priority foreign countries" and investigated for possible trade sanctions under the "Special 301" section of the Trade Act, 19 USC 2411.²⁹ The biggest offenders on the 1994 list were China, Argentina, and India, but particularly China. Robert Holleyman, president of the Business Software Alliance, indicated that the U.S. software industry lost \$322 million in China in 1993, adding that 94 percent of the packaged software in China is thought to be pirated.³⁰

²⁸"Copyright Term of Protection - Harmonized with European Union." LEXIS-NEXIS Hot Topics Intellectual Property Law 4 Aug. 1995.

²⁹"Industry Urges Firm U.S. Position." Patent, Trademark & Copyright Journal 30 Jun. 1994: 217.

³⁰"Industry Urges Firm U.S. Position." Patent, Trademark & Copyright Journal 30 Jun. 1994: 217.

On April 29 of last year, Japan was placed on the 1995 priority list. This leaves it open to possible targeting in the future as a Special 301 priority country. This switch was due in part to then current worries in the White House as well as in some American business circles that Tokyo might amend its copyright laws to allow decompilation or reverse engineering of computer programs. It also reflected mounting U.S. questions about Japan's willingness to crack down on software piracy and increasing domestic industry concerns about the competitive impact of that country's patent system. In response, the Japanese government promised to limit the time between the initial submission of a patent application and its acceptance or rejection to three years versus the typical five or six years, a major competitive handicap for high technology firms with their very short product life cycles. These and other changes notwithstanding, Clinton administration officials argue that Japan's patent system still works to the disadvantage of innovative American firms.³¹

CAN INTELLECTUAL PROPERTY LAWS PROVIDE SECURITY?

As discussed in the previous sections, there are a variety of ways to provide security for computer objects. One must bear in mind that no means of security is fail safe, and therefore additional precautions must be taken to assure the highest possible protection.

We have seen that for hardware, probably the most appropriate form of intellectual property protection is the patent. Hardware is, in a sense, an invention, and therefore may be patented.

Patents are not typically encouraged for software because software is seen as the representation of an algorithm or idea, which is not by definition, subject to a patent. Conversely, trade secret protection applies well to computer software because it protects the secrecy of the algorithm (the code), while still allowing distribution of the executable program. However, because trade secret protection does not protect against copying the program (i.e. source code), it does not apply any penalty to software pirates who copy the program and then sell it for commercial gain. Thus, copyright protection could be the most appropriate means of software protection.

Finally, protecting the data in a database is more of a gray area because data in and of itself is not patentable. Trade secret protection does not really seem appropriate because most of the time the underlying data is not secret. The remaining type of intellectual property protection currently on the law books is copyright, but as mentioned in the case of the white pages of a phone directory, a database must have some element of originality and creativity for its material to be copyrightable. Many databases may not meet these requirements, and therefore not fall under the protection of any of the available intellectual property laws.

Over time, many of these fuzzy areas of the law will become clear as additional legislation is made and additional court cases are heard. A hope of all those involved in this type of litigation is that the court systems will take a proactive stance on clarifying these issues. With the short life cycle of most of the hardware and software on the market today, a court decision could be yesterday's news before it is even decided.

TRENDS FOR THE FUTURE

Recently, the Supreme Court brought to a close a decade long search in court for the proper and best way to protect software. *Lotus Development Corp. v. Borland International*, the first case to reach the Supreme Court addressing the issue of copyright protection for computer software. At issue was whether the language used in command menus developed by Lotus for its 1-2-3 software program was a copyrightable work as defined by the Copyright Act. Lotus argued that the software's command menu was a literary work entitled to the same copyright protection as other literary works. Borland countered that the words in the command menu were more like basic English grammar and should not be afforded copyright protection.³² The Supreme Court's 4-4 tie vote affirmed the First Circuit's decision that Borland's copying the menu tree from Lotus' 1-2-3 spreadsheet program was not a copyright violation. But the unwritten decision has no precedential value and will give scant guidance on what aspects of software are copyrightable expression and what are uncopyrightable methods of operation. The 4-4 tie in the *Lotus v. Borland* case will likely encourage others to look to other forums.

In lieu of Supreme Court guidance, technology lawyers will have to turn to more than a half dozen similar software copyright cases pending in Federal courts around the country to determine the boundaries of protection. Nine copyright experts interviewed recently say that recent court decisions have leaned toward favoring defendants in software copyright disputes over command menus. They also say the courts appear to be recognizing the legitimacy of copying to achieve compatibility between computer programs. When the First Circuit Court ruled last year that the command menu of Lotus' 1-2-3 spreadsheet program, was not protected under copyright law, it reasoned that the menu was a "method of operation" rather than an expression of an idea. In other words, the 1-2-3 user interface was merely functional. Under Section 102(a) of Federal copyright law, computer programs appear to have protection as "literary works" which include works expressed in "numerical symbols" and "embodied" in "disks" or "cards." But another provision, Section 102(b) bars copyright protection for any "idea...process [or] method of

³¹05/26/95, Japan Economic Institute of America JEI Report, Section no 20 Vol. 199, Washington Watchful on Tokyo's Protection of Intellectual Property Rights

³²LEXIS-NEXIS Hot Topics Intellectual Property Law, Jan. 11 1996

operation." The tension between those two provisions is at the heart of the debate over whether protection should be afforded to a program's user interfaces as opposed to its underlying code, which is generally considered to be within the scope of copyright law.³³

Other trends in the intellectual property and computer software arena include a multi-year plan to revise the Uniform Commercial Code (UCC) in an effort to give more credibility to shrink wrap licenses. Companies have had limited success in court enforcing software licenses because the buyer is unable to bargain over terms and because some courts have found that the contracts are an impermissible trick of the Copyright Act.³⁴

Many companies are putting greater emphasis on protecting their products through patents rather than by or as a complement to, copyrights. This shift recognizes that courts have been loosening their restrictions over patents while tightening requirements for copyrights, and that patents have a greater certainty than copyright. It's generally easier to interpret a patent claim than to agree on copyright's coverage in software.

Another trend that appears to be emerging is the acceptance of copying for the sake of computability, or interoperability between computer programs. Jonathan Band, a copyright lawyer in Morrison & Foerster's Washington, D.C. office says that courts have come to recognize that compatibility between products has become a pervasive aspect of the computer industry. With a spreadsheet program, the operating system it is designed for is nearly as important as what the software's basic purpose is. In its battle with Lotus, Borland argued that in part its adoption of the 1-2-3 menu command in its Quattro Pro spreadsheet sheet was crucial to obtaining compatibility with macros. However defendants should not be able to simply invoke compatibility as a pat defense against claims of infringement without showing functional constraints on how a program is defined for developing application software to be compatible with an industry standard.³⁵

CONCLUSIONS

This paper has attempted to show that the three goals of computer security, that is, secrecy, availability, and integrity, may allow various computer objects, specifically computer software, to be afforded some protection under the law in addition to the protection offered by physical devices and procedural methods. The security devices used by the law are the patent, the copyright and the trade secret. A discussion of the nature of each of these legal devices was presented, along with a brief legislative history. Recent legislation was presented concurrently with a discussion of the economic aspects of computer software piracy and the need for more stringent penalties. The legislation and the economic aspects are inseparable because one literally drives the other. In addition, an overview of the international arena was presented with the corresponding intellectual property laws in various nations. Finally, trends for the future of intellectual property rights and computer software were presented.

I believe that in recent years many advances have been made to address the challenges being faced by this industry. It is said that half the battle is recognizing and defining the problem. Unfortunately, however, this is not enough. In order to meet the future head on, the present is going to need a head start.

BIBLIOGRAPHY

1. Kantrowitz, Barbara. "My Info is NOT Your Info" Newsweek July 1994: 54.
2. Pfleeger, Charles P. Security in Computing. New Jersey; Prentice Hall PTR, 1989.
3. Spanner, Robert A. "The Brave New World of Criminal Software Infringement Prosecutions." The Computer Lawyer November 1995, Page 1.
4. Walsh, Mark. "Copyright Experts Scan Horizon for Rules on Software." The Recorder Feb. 1996.
5. Congress, Office of Technology Assessment, *Finding a Balance: Computer Software, Intellectual Property, and the Challenge of Technological Change*, OTA-TCT-527 (Washington, DC: U.S. Government Printing Office, May 1992).
6. Department of Commerce, Patent and Trademark Office, *Basic Facts about Patents*.
7. Patent Trademark and Copyright Journal 18 Apr. 1991.
8. Patent Trademark and Copyright Journal 11 Jun. 1992.
9. "Senate Passes Bill to Stiffen Criminal Penalties for Software Infringement." Patent Trademark and Copyright Journal 4 Jun. 1992: 121.
10. "Legislation, Copyrights." Patent Trademark and Copyright Journal 9 Apr. 1992: 511.
11. "House Panel Amends Bill on Criminal Penalties for Software Infringement." Patent Trademark and Copyright Journal 17 Sept. 1992: 485.

³³Walsh, Mark. "Copyright Experts Scan Horizon for Rules on Software." The Recorder Feb. 1996.

³⁴"End of an Era, Software Wars Are Over." Information Law Alert: A Voorhees Report, 26 Jan. 1996.

³⁵Walsh, Mark. "Copyright Experts Scan Horizon for Rules on Software." The Recorder Feb. 1996.

12. "Bill Would Strengthen Penalties for Criminal Infringement on Internet." Patent, Trademark and Copyright Journal 10 Aug. 1995: 368.
13. "Industry Urges Firm U.S. Position." Patent, Trademark & Copyright Journal 30 Jun. 1994: 217.
14. United States. Information Infrastructure Task Force, National Information Infrastructure Report, 172 D 128, 1995.
15. "Free Speech and Copyright in Cyberspace: Legal Issues Surrounding the Internet." On-line Libraries and Microcomputers Mar. 1995.
16. Full Text of Bills, 104th Congress, 1st session, H.R. 2441, September 29, 1995.
17. "Prepared Testimony of Barbara A. Munder..." Federal News Service 7 Feb. 1996.
18. "Copyright Term of Protection - Harmonized with European Union." LEXIS-NEXIS Hot Topics Intellectual Property Law 4 Aug. 1995.
19. LEXIS-NEXIS Hot Topics Intellectual Property Law, Jan. 11 1996
20. Japan Economic Institute of America JEI Report, Section no 20 Vol. 199, Washington Watchful on Tokyo's Protection of Intellectual Property Rights
21. "End of an Era, Software Wars Are Over." Information Law Alert: A Voorhees Report, 26 Jan. 1996.

CASE STUDY OF INDUSTRIAL ESPIONAGE THROUGH SOCIAL ENGINEERING

Ira S. Winkler

National Computer Security Association

10 South Courthouse Avenue

Carlisle, Pennsylvania 17013

winkler@ncsa.com

(717) 258-1816 x257

Abstract

The Federal Bureau of Investigation estimates that U.S. Corporations lose \$100 Billion annually due to industrial espionage. While many people believe that the espionage is committed by well financed organizations that can only be stopped by national agencies, that is very incorrect. Industrial espionage usually exploits simple and very preventable vulnerabilities to produce tremendous results. By focusing on comprehensive security, and not just technical security, information security professionals can significantly hamper adversary attempts to steal their organization's information assets. The presentation that describes this paper presents a case study of an actual industrial espionage attack against a large U.S. corporation.

Introduction

The theft of sensitive information from U.S. corporations is the goal for many foreign nations and companies. Adversaries do not care about what form the information takes. Whether information is in electronic format or is thrown away in the trash, it is irrelevant as long as the information is compromised. Unfortunately for most corporate security programs, there is a preoccupation with technical security that leaves information very vulnerable to basic espionage methods.

Information security professionals focus their efforts on what they know best. When they allocate their limited budgets, the division of funds reflects their perceived needs, which are basically technical security mechanisms. Firewalls and other Internet security mechanisms are the hottest selling products. While firewalls go a long way in preventing the traditional computer hackers from intruding into a corporate computer network, they do nothing to stop the most significant source of computer crime: Insiders. Two recent studies show that insiders were responsible for more than 70% of information related thefts [1, 3]. The threat prevented by firewalls is minimal, because a focused attack will bypass the strongest protection mechanisms.

Information comes in many forms, and must be protected in all of its' forms. Information security is not computer security. While computer security is an integral part of a good security program, it is only a part. Comprehensive security includes physical, personnel, operational and technical security. Industrial spies know how to bypass any strong part of a security program to attack an organization at its' weakest point.

Industrial Espionage Methods

There are wide spread reports of former Soviet Bloc intelligence operatives acting as freelancers to the highest bidders, as well as foreign intelligence agencies refocusing their efforts on U.S. companies as opposed to the U.S. Government [4, 5, 7]. These intelligence organizations bring their tried and true methods with them. Unfortunately, most corporate security managers are not aware of the threats and the methods they employ. Intelligence gathering methods are more effective on companies than they are on governments, because companies do not have the appropriate countermeasures in place.

Legal Methods

There are several forms of industrial espionage that are legal. These methods include the purchase of companies or products, and has the net result of transferring technology to the previous competitor. There are many examples of foreign firms buying U.S. companies to acquire critical technologies. The threat to U.S. national competitiveness is very serious, however there is little that can be done by a corporate security manager to prevent this type of information acquisition [7].

Another legal method of acquiring U.S. technology involves pressuring companies into giving up their technology. Basically, this involves the blackmail of U.S. companies by foreign countries. In order for a company to do business in the foreign country, the company must train native workers in a critical technology. It is then up to the company to decide if the cost of doing business with the country is worth it. At this time, a corporate security manager may or may not be involved in the decision. Obviously from an information protection perspective, the answer is obvious. From a business perspective, it is much less clear [7].

The practice of joint ventures with competitors also provides a huge opportunity for U.S. companies to give up sensitive information. During the process of expanding the state-of-the-art, a company must divulge its' knowledge of the state-of-the-art [2, 7]. In some cases, a joint venture may be the only method for a company to enter a foreign market. Again, there is a Cost/Benefit Analysis to be performed prior to entering into such a venture.

Open source information (OSI) also provides a wealth of knowledge for industrial competitors. OSI takes a variety of forms including newspaper articles, corporate Annual Reports, patent filings, court papers, and marketing information. For example, most requests to review patent filings are by foreign nationals and third party research firms. By reviewing OSI, competitors can determine a tremendous amount of information about a company and their products. The losses are tremendous and unfortunate, especially when a company does not realize that they are giving away all of their information [7].

The hiring away of employees also results in the transfer of knowledge to a competitor. While many former employees do not intend to divulge sensitive information, the transfer of the knowledge is inevitable. In the performance of day to day activities, it is impossible not to take into account the knowledge that a person has developed. For example, if a person is trying to

price a job for their present company, it is impossible for them not to consider the pricing structure of their former employer, who is now competing for the same job.

Many companies use trade shows and conferences to elicit information from competitors. Typically, corporations send their researchers and marketing staff to these events to either stay abreast of the latest research or sell services or products. These people usually give out information better than they collect it. Companies involved with industrial espionage also send information collection specialists to these events. They usually act like potential customers or fellow researchers to elicit information from people that are all too willing to give it up. Through advanced training these collection specialists have perfected the art of drawing out as much information as possible [7].

Foreign countries make it a habit to contact natives of their country that have had contact with a targeted company. These natives are requested to divulge information that they have obtained from the company. It is typical for individuals to have more loyalty to their native land as opposed to a foreign company that they have worked for. These people are readily recruited by foreign intelligence services, and the knowledge that they divulge is quickly passed to foreign companies and countries. In some cases a foreign intelligence service may recruit a national to work for a U.S. firm. They will assist in obtaining a job for that person, and help in any way possible. The individual may not realize that they will be contacted at a later time to compromise information [7].

Illegal Methods

Many of the previous methods appear to border on criminal activity. It is a fine line between a foreigner divulging information to their native country and a U.S. citizen selling the information to that country.

Many industrial espionage cases involve the use of insiders to steal information. The cooperation of insiders can occur in many ways, depending on the circumstances. As with traditional espionage cases, the recruitment of moles is frequently used. Moles are employees of a targeted company, or someone with access to the company, that agree to cooperate with the criminals, usually in exchange for money. These people abuse the access that they have to steal information, or possibly just hand over information that they already have access to. They are well established within the target, and can typically move through the organization unchecked. Moles may be recruited by the industrial espionage organization, or may volunteer their services. It is not unheard of for people to approach their company's competitors to sell corporate secrets [1, 4].

The recruitment of a mole can be risky for an attacker, because there is the possibility that the potential mole might report an initial approach to corporate security personnel. For this reason, it is very likely that an industrial spy will attain their own position within the target. Large companies have an on-going recruitment process, and it is easy for spies to obtain a job. Once inside the company, they can abuse their access and usually go undetected in their thefts of information. Again, most companies are in the process of increasing their perimeter security mechanisms, but leave their internal system without protection [4].

There are less sophisticated, but still effective methods for stealing information. Espionage could involve breaking into buildings and offices to steal the desired information. Industrial spies will go through locked and unlocked office spaces, search file cabinets, examine unprotected computer systems, etc. If a person knows where the targeted information is located, it could be extremely profitable for them to commit a simple break in. Spies will also go through trash dumpsters and other garbage containers to gather information. While many people think that this is ridiculous, it is extremely effective [4, 8].

If a company has people that travel frequently, it is very possible that their travelers could be the subject of sophisticated surveillance efforts. U.S. executives have reported that their hotel rooms appear to have been searched, that their telephone calls have been monitored, etc [7]. The value of the information that they know, ultimately drives their risk of being watched by adversary organizations.

I have left the discussion of technical collection methods for last, not because it is unimportant, but because the focus on technical countermeasures causes major security vulnerabilities with regards to the other information security disciplines. Industrial spies can collect information by computer hacking, tapping telephones, sophisticated cryptanalysis efforts, etc. There should be dozens of other papers at this conference describing technical intrusion methods in detail. Industrial spies use all known methods of technical information collection. Due to the effectiveness of currently known methods, it is unlikely that they have to develop any new methods.

Clearly, computer intrusions can yield a tremendous amount of sensitive information, however it is the goal of this paper to stress that it does not matter how much information an industrial spy ring obtains, but what information they obtain. A single document can be worth billions of dollars, and it does not matter if the information is found in a computer or in the garbage [4, 8]. In many cases acquiring terabytes of data can hinder the collection of a single document, because of the difficulty of data reduction.

Preventing Industrial Espionage

Since the methods used by industrial spies are the same as those used by traditional spies, the countermeasures used to prevent traditional espionage can prevent industrial espionage [7]. There is a great deal that commercial organizations can learn from Department of Defense security practices. While I am not advocating total adherence to DoD standards, companies must employ a level of countermeasures that are justified by the potential losses that the company can suffer. For many firms, the potential losses can easily be valued in the billions of dollars. Information security efforts must therefore address comprehensive countermeasures, that are as comprehensive as the methods employed against them. There are four parts of a comprehensive security effort that enhance and support each other: Technical, Operational, Physical, and Personnel Security. This paper introduces the concept of comprehensive security. It is strongly recommended that other papers follow up on the following concepts.

Technical Security

Technical security countermeasures reduce the vulnerabilities present in electronic systems. As many other papers at this conference address, countermeasures ensure the confidentiality, integrity, and availability of computer systems and networks. A good technical security effort also protects other electronic systems such as voice mail. The technical issues are well known and are satisfactorily addressed elsewhere.

Operational Security

Operational security addresses the business processes in use by a company that could compromise information through non-technical means. For example, the DoD policy concerning information access only on a "Need to Know" basis helps prevent the unnecessary proliferation of information. Likewise, policies on restricting the use of open communication lines, such as the Internet and telephone systems, reduces the potential for the compromise of information. Other operational security issues include enforcing your own security policies on your vendors and suppliers. It would make no sense to perform background checks on your own employees, while contractor employees, who have free access to your facilities, go unchecked.

Operational security is a complicated issue, and requires a thorough study of the way a company does business. This includes the marketing process, which presents a major vulnerability due to the exuberance a sales people trying to close a deal by offering sensitive information. Companies must examine the entire research, development, manufacturing, and sales process for potential ways that information could be compromised. There must be a clear understanding of who to disclose information to, and under what conditions and controls.

A strong security awareness program is the foundation for a strong operational security program. People must know what information they should protect, and specifically how to protect it. Everyone should be encouraged to report any questionable circumstances, and know who to report it to. Security managers cannot assume that security issues are common sense when there is no baseline for common knowledge. Operational security issues must be further elaborated and studied in other forums.

Physical Security

As previously discussed, a large number of information compromises occur due to simple breaking and entering, and theft. Physical access to facilities should be carefully regulated and controlled. This includes limiting the access of visitors and contractors, as well as your own employees. Nobody should have a free roam of all corporate facilities.

All employees must wear access badges that indicate their status, such as employee, temporary, visitor, or contractor. This feature helps to reduce the threat of people overstating their authority. Obviously, there should be an operational security policy that encourages all people to look at badges. Another physical security issue to be addressed is the control of garbage. There have been numerous incidents of serious information compromises that have occurred solely from the

content of an organization's garbage. The U.S. military has several units devoted to trash intelligence, and invests millions of dollars in the proper disposal of classified waste. Companies that have very high value information must also consider the control of their garbage.

Security programs must also stress the use of available protection mechanisms. Locks on office doors and file cabinets frequently go unused in many organizations. Clean desk policies, that require all sensitive information to be locked up, must also be enforced. There are also computer locking products available that prevent computer access if it is turned off or idle for a certain period of time. These products prevent the exploitation of computers that are not properly turned off when not in use.

Personnel Security

There must be a thorough investigation of all people with potential access to sensitive information. Since most information might be sensitive to different departments within an organization, it should probably be a blanket policy to have a background check performed on all employees. The term employees is used broadly to include anyone with physical access to facilities or information. Facilities include any computer terminal that has access to corporate information.

Many organizations do not consider the access and opportunities that seemingly minor employees, such as janitors, clerical workers, and security guards, have to steal information. A recent edition of *2600: The Hacker's Quarterly* had an article on how to obtain a job as a janitor [6]. Criminal elements understand the potential of low level positions, and it is time for security managers to address that potential.

Systems administration staff should also establish a strategic relationship with the Human Resources department. It is critical to be aware of any pending employee departures that could be under less than amicable circumstances. Also, systems administrators must lock the accounts of departed employees on the day that they leave the company.

Case Study

The case study for the presentation addresses a penetration test performed against a large high technology firm at their request. The goal of the test was to simulate an industrial espionage attack, within the funding parameters. A comprehensive attack strategy was used to simulate an attack as accurately as possible. The attack included the use of Open Source Research, obtaining a position as a temporary employee within the target, misrepresentation of responsibilities by the temporary, abuse of physical access, internal hacking, internal coordination and facilitation of external hackers, and straight external hacking.

The results were staggering. Within one day of the on-site activities, over \$1,000,000,000 of information was "stolen." While the firewall was impenetrable and Smart Cards prevented access from outsiders, information was compromised almost at will by an insider. This was accomplished in a company that has a tremendous technical security program. The security

manager understands their vulnerabilities, and wanted an independent assessment of the vulnerabilities to demonstrate the seriousness of the problem. A detailed description of the case study will be presented.

Conclusions

There is a tremendous focus by information security professionals on technical security. This is probably due to the traditional background of information security professionals being from a technical background. When they receive funding for their efforts, their initial reactions are to spend the money on what they are most familiar with, which usually does not include awareness programs or the acquisition of shredders. Firewalls and other security tools are important, but unfortunately they only address a small part of the problem. All recent studies show that insiders pose the most serious threat to information, and firewalls do little to prevent the abuse.

It is time for commercial information security professionals to realize that information security is more than computer security. A comprehensive security program that includes all security disciplines is the only effective countermeasure to a coordinated industrial espionage attack. A determined attacker will exploit the most vulnerable access points, and will not stop trying until they get what they want or are caught. A detailed and continual awareness program is the best method to deter many attacks. If all employees know what to look for, then the chances for the attack to be successful are minimized.

Bibliography

1. American Society for Industrial Security (1996), *Study on the Theft of Proprietary Information*, Arlington, VA: ASIS.
2. Cox, J. (1996), Siphoning U.S. Companies' Knowledge, *USA Today*, February 16, p. B1.
3. Katz, A. (1995), *Computers: The Changing Face of Criminality*, Unpublished dissertation: Michigan State University.
4. Pasternak, G. (1996), The Lure of the Steal, *U.S. News & World Report*, March 4, p. 45.
5. Schweizer, P. (1993), *Friendly Spies*, New York: Atlantic Monthly Press.
6. Voyager (1994), Janitor Privileges, *2600: The Hackers' Quarterly*, 11(4).
7. White House (1995), *Annual Report to Congress on Foreign Economic Collection and Industrial Espionage*, Washington, DC: Government Printing Office.
8. Winkler, I. (1996), Assignment Espionage, *InfoSecurity News*, 7(3), p26.

LEGAL ASPECTS OF ICE-PICK TESTING

Dr. Bruce C. Gabrielson
Kaman Sciences Corp.
Alexandria, VA

in Association with Naval Research Laboratory
Contract No: M00014-93-C-2033

Abstract

The Ice-Pick package is a window driven program that provides a multi-layered approach to network testing. The automated tool is used to identify frequently exploited security problems present on well known UNIX based operating systems. Information provided by testing is used to determine what protective mechanisms need to be implemented by network administrators.

The paper deals with two issues of primary concern, the user's legal basis for performing vulnerability identification testing, and the consequences of unauthorized use or release of the software itself. It is essential for self protection that the tester understands what he or she can legally do with a tool such as Ice-Pick. The issue of trust can also effect users. Trusting each user to protect Ice-Pick against unauthorized release is essential for absolute control of the technology involved.

The structure of this document allows traceability from top level law through applicable Navy regulation. The most important points are the understanding of what monitoring involves, and knowing what the Ice-Pick test tool can be used for. The use of other penetration type testing tools, such as SATAN, will not be discussed, nor will the regulatory requirements of non-Navy organizations. However, the discussion can be applied to using similar test tools in other organizations.

Introduction

This paper discusses the legal basis for performing Ice-Pick testing in the Navy, and the consequences of unauthorized use or release of the software itself. It is essential for self protection that the tester understands what he or she can and can't do with the tool. Providing the information background for the tester to evaluate test activities is one means of accomplishing affective conditioning. Therefore, the legal basis supporting testing and accountability when using the tool will be derived first.

Trusting the user is another issue. Although trust of each user against the unauthorized release of Ice-Pick is assumed, its distribution must be absolutely controlled. Therefore, a discussion of the repercussions of improper release, particularly to the user, will enhance the user's awareness of the problem, as well as provide the legal basis for prosecution should the software find its way into the wrong hands.

Background on Ice-Pick

Ice-Pick is an unclassified automated tool that can be used for testing network vulnerability profiles. The Navy developed it to proactively attack its own networks for SST&E purposes. Ice-Pick does what it is intended to do very well. The Ice-Pick user can only test for vulnerabilities. Private information can not be accessed with the Ice-Pick application running.

Ice-Pick's software incorporates protection mechanisms to ensure only pre-authorized sites will be targeted. The software can also be directed to run on only one pre-designated machine. However, these controls are directed at software operation. Using the program requires a certain level of technical skills. The skills required are information sensitive in nature in that the individual using the program could basically become an accomplished "hacker".

The problem with the deployment of a proactive test tool is that it is capable of being used both for and against a network. Ice-Pick is simply a tool which has a number of internal program safeguards, and also needs a certain level of expertise to be used properly. Since it relies on applying technologies that could be misused, the tester needs to fully understand both regulation and capability in order to correctly apply tests where they may be legally be used.

General Legal Policy

Formal adherence to detailed security standards for electronic information processing systems are necessary for industry and government survival. These security standards are necessary because of the amount of information, the value of the information, and ease with which the information can be manipulated or moved. However, standards must be backed by law if they are going to be mandated. Government organizations are required to comply with these laws, as well as comply with numerous regulations related to unclassified sensitive and classified environments. Each organization has, therefore, developed its own set of instructions regarding how it will comply with top level laws and requirements.

Top Level Legal Traceability Issues

Two federal laws drive the need for protecting an organization's network and computing resources. The National Computer Security Act requires computer security implementation and training on Government computers in order to provide for information protection. The second law, the Privacy Act, protects private information on individuals. Government organizations should be in full compliance with these and other security or privacy type regulations. In addition, Department of Defense organizations have issued site specific instructions regarding the protection of their sensitive, but unclassified information. Penalties for the unauthorized release of protected information, as well as specific access authorization criteria are well documented.

There is also a personal liability issue. Down time to get an organization's network back on-line, or to simply recover data after a virus attack can be very expensive. Costs can also be high if certain types of data is manipulated to show other than actual information. Therefore, it is important for the tester to understand that unauthorized use of any software for the purpose of manipulating or otherwise destroying data can result in personal legal responsibility for organizational financial loss.

Privacy Act and Federal/Public Law

The top level Federal Statute relating to private information of an individual citizen is covered under the Privacy Act of 1974. This law protects individuals from disclosure of various categories of information, and has significant penalties imposed on violators. A important provision of the Act is shown below:

Privacy Act of 1974 (as of Jan 1993)

552a. Records maintained on individuals

(b) Conditions of disclosure.--No agency shall disclose any record which is contained in a system of records by any means of communication to any person, or to another agency, except pursuant to a written request by, or with the prior written consent of, the individual to whom the record pertains, unless disclosure of the record would be-- (1) to those officers and employees of the agency which maintains the record who have a need for the record in the performance of their duties;

Two other laws have a direct bearing on those who are responsible for protecting computer assets.

Public Law 100-235 (Computer Security Act) is intended: "To provide for a computer standards program within the National Bureau of Standards, to provide for Government-wide computer security, and to provide for the training in security matters of persons who are involved in the management, operation, and use of Federal computer systems, and for other purposes."

OMB Circular A130 Federal ADP guidelines. "The Paperwork Reduction Act (44 U.S.C. Chapter 35) assigns the Director of the Office of Management and Budget (OMB) responsibility for maintaining a comprehensive set of information resources management policies and for promoting the application of information technology to improve the use and dissemination of information by Federal agencies."

Network Monitoring and Privacy

How are privacy and network monitoring related? When dealing with a computer tool, several items are considered. For example, will using the tool result in keystroke monitoring or packet detection, or will it allow real-time communications detection. Related to electronic monitoring, privacy rights are found in the Electronic Communications

Privacy Act of 1986, and are embedded in the US Constitution. The Electronic Communications Privacy Act of 1986 (ECPA) provides additional privacy protection against monitoring. Title I of the ECPA includes electronic communications and its protection. Title II of the statute protects stored communications. The Fourth Amendment of the Constitution provides that:

"the right of the people to be secure in their persons, houses, papers, and effects, against unreasonable searches and seizures, shall not be violated; and no warrants shall issue, but upon probable cause,..."

As indicated, compromising one's privacy is a serious issue, requiring both a formal process and probable cause. In other words, legal action is necessary to compromise an individual's privacy.

Accessing Stored Communications

Both real time and stored communications could be considered private. Section 2701 of Title 18 of United States Code makes it a criminal offense to unlawfully access stored communications. It is a violation of this section to intentionally access without authorization a facility through which an electronic communication service is provided; or to intentionally exceed an authorization to access that facility and thereby obtain, alter, or prevent authorized access to a wired or electronic communication while it is in electronic storage in such systems. This is a criminal statute and fines and imprisonment can result.

If an individual has a reasonable expectation of privacy in his or her computer (hardware or software), there must be some legal safeguards put in place before a search and seizure of the computer or communications can take place. If the action is part of a

Relevant Laws/Acts/Circular

PL 97-255

Federal Managers Financial Integrity Act of 1987

PL 99-473

Comprehensive Crime Control Act of 1984

PL 99-474

Computer Fraud & Abuse Act of 1986

PL 100-235

Computer Security Act of 1987

PL 100-503

Computer Matching and Privacy Protection Act

OMB Circular A-130

Mgt. of Federal Information Resources

OMB Circular A-123 & 127

Internal Control/Financial Management Systems

Other Relevant Documents

OMB Bulletin 89-22

OMB Bulletin 90-08

EO 12333

EO 12356

DCI DIR 1/16

US CODE, TITLE 18, SECTION 2511

Applicable Defense Statutes (Navy Example)

DOD 5200.28-STD (Orange Book)

OPNAVINST 5239

SECNAVINST 5239

SITE INSTRUCTION

criminal investigation, then a warrant is required. Note that even in situations where government employers or supervisors seek access to an employee's computer (or office, desk, etc.) there must be, in the absence of a warrant, a reasonableness determination and a balancing of the employee's privacy interests that will withstand judicial scrutiny. Determining what level of constitutional protection a government employee has in a work-setting depends on the circumstances and whether the employee has a reasonable expectation of privacy.

On the issue of reasonableness, one issue of privacy relates to the practice of network monitoring by individual Government organizations. Neither a warrant nor a reasonableness determination is required where there is no reasonable expectation of privacy, or where the individual has consented to intrusion. Within the Department of Defense, all DoD interest computer systems and related equipment are intended for the communication, transmission, processing, and storage of official US Government authorized (and owned) information only. US Government telecommunications systems and information systems (ISs) are subject to periodic security testing and monitoring without prior notification to ensure proper functioning of equipment and systems including security devices, to prevent unauthorized use and violations of statutes and security regulations, to deter criminal activity, and for other similar purposes. Use of any Government network or equipment constitute consent to monitoring.

Monitoring notices indicating that there is no right to privacy in the system by any user is advantageous relative to reasonableness. Some Government agencies (such as the Navy) have complete control over their network and include a monitoring notice such as that shown below which appears every time a user logs onto many networks.

"All Department of Defense telecommunications and automated information systems are for the communication, transmission, processing, and storage of U.S. Government information only. The systems and equipment are subject to authorized monitoring to ensure proper functioning, to protect against unauthorized use, and to verify the presence and performance of applicable security features. Such monitoring may result in the acquisition, recording, and analysis of all data being communicated, transmitted, processed, or stored in this system by a user. If monitoring reveals possible evidence of criminal activity, such evidence may be provided to law enforcement personnel. Anyone using this system expressly consents to such monitoring."

Unfortunately, implied consent isn't always accepted by an employee. In addition, not every organization can claim they have the legal right to gain access to an individual's personal files. Since testing may result in the identification of an access point, one of the initial concerns a testing organization has is their legal basis for testing.

Lets examine closely what a penetration test tool really does. Remember that these tools work by actually attacking a network. If the attack is successful, it can also be used as an initial step in the monitoring process. Public Law 99-474 applies to those who knowingly access a computer without authorization, or to those who exceed their authorization. Additionally, although the site users are normally pre-warned, the actual

testing of a particular user's machine must be accomplished with sensitivity to both the user and the system manager responsible for the network being tested to avoid any misunderstandings.

There also may be site/organization specific legal issues in accessing sensitive non-classified information which may include private information. However, informed consent of the user (the login banner) minimizes legal issues presented to the system administrator by using tools such as Ice-Pick. An organization should not perform network testing until it can certify that 100% of the computers to be tested display the proper monitor banner. Additionally, some system administrators choose to use a formal users agreement which lays out the same type of information contained in the banner, and contains the user's signature acknowledging an understanding of the banner.

In spite of the implied consent provided by the use of login banners, understand that formal computer monitoring is allowed only in very limited situations and only when pre-approved at the appropriate level. For the Department of Defense, Communications Security (COMSEC) monitoring is under the cognizance of the National Security Agency, who then delegates to service cryptological elements.

Use Within the DoD

The Computer Security Act established the guidelines and rules for the protection of Government computing assets. Within the Department of Defense (DoD), security rules have been established to implement the Computer Security Act and protect computer systems which process classified or sensitive but unclassified information. These rules are intended to provide guidance for both manufacturers and for users. Computers that meet the National Computer Security Center's (NCSC's) trusting criteria have integrated safeguards into their operation such that only the users "trusted" to have access to the restricted data can actually gain access.

The rules are described in a series of documents known as the Rainbow Series. Currently there are six levels of Trusted Computer classifications as described in the Orange Book¹. Requirements for software/hardware security policy, accountability, assurance, and documentation vary depending on the level of security to be achieved.

From the initial Rainbow Series documents, various DoD organizations established and developed their own programs to implement information security rules.

Navy Regulations/Instructions

The Navy's computer security program structure followed the guidelines established by DoD 5200.28, plus has incorporated the requirements of newer laws and directives, including the Privacy Act. The Navy's current program is based on the requirements of

¹*Trusted Computer System Evaluation Criteria, DoD "Orange Book", DOD 5200.28, DECEMBER 1985.*

SECNAVINST 5239.3 dated 14 July 1995. Policy will be further implemented by the OPNAVINST 5239.X currently in draft form. Specific to the type of protection addressed by Ice-Pick testing, the following paragraphs relate directly, with bold type indicating the specific wording:

SECNAVINST 5239.3

"7. Policy"

"b. Fundamental INFOSEC Policy"

"(1) Data processed, stored and transmitted by information systems shall be adequately protected with respect to requirements for confidentiality, integrity, availability and privacy."

"(2) The nature of the DON mission, accompanied by connectivity and data aggregation issues, has led to the determination that all unclassified information processed by DON information systems is sensitive. Therefore, all DON information systems shall be protected by the continuous employment of appropriate safeguards."

IS Security Program Implementation

The Information System (IS) Security Program developed by Government organizations is designed to provide end-users with good security practices as well as comply with current Government requirements. This practice establishes good habits within the local community and narrows the possibility of: disclosure of data, equipment loss, and misuse of government resources.

The Navy's IS Security Program is designed to ensure the confidentiality, integrity, and availability of its computing assets. It is driven by a primary need. The need to maintain configuration management controls over equipment that may be susceptible to identified threats.

The potential risks to Navy computers posed by potential threats establishes the basis for controlling the configuration management of all IS which process classified and unclassified but sensitive information. The Navy has chosen to address this control need through the establishment of a Risk Management Program, which in turn requires a verification process to ensure its viability.

The ultimate recognition of the potential hacker/cracker threat beyond the stand-alone IS has resulted in an expansion of the risk management program as well as the implementation of a network oriented security system testing & evaluation (SST&E) program.

Navy networks are constantly bombarded by off-site hacker/cracker penetration attempts. In the Navy's network monitor and test role, an active evaluation, test, and

continual upgrade of network security protection measures are necessary throughout the IS's life cycle.

Security System Test and Evaluation

The SST&E function is the active auditing part of the Navy's IS security configuration management procedure. SST&Es gather empirical data on individual systems and are examined by the DAA in the evaluation procedure. Applying the SST&E process to the active testing of networked ISs provides the local IS Security Group with the ability to protect Government computing resources under its control. The process evaluates the effectiveness of in-place countermeasures against incidents that would effect the networked IS in a negative manner. If the in-place countermeasures are inadequate, the SST&E will uncover this fact so they can then be rectified.

SPAWAR Security Program Compliance

Within the Navy, the Chief of Naval Operations (CNO) has appointed the Director, Space and Electronic Warfare, as the Navy's Senior Information Systems Security Manager (SISSM). Among the SISSM's tasks are maintaining the OPNAVINST 5239.X and its supplements, and maximizing the use of automated security related tools. As the following document describes, Ice-Pick is considered by name as one of these tools.

Automation in Certification and Accreditation, SPAWAR PD 51, Section 2.0 Automation Support for the Naval Systems Security Engineering Process.

"2.1.6 Secure System Operation"

"The system security personnel must be able to maintain and monitor system operation and determine the security effectiveness of the installed system. During operation, the system security personnel need to be able to probe the system, control system access and usage, and understand the impact of system configuration changes to the system security."

"2.3 AUTOMATION OBJECTIVES"

"Specific activities within the security engineering process that are suited to automation are listed below:"

** Conduct security testing (i.e., Certification Test & Evaluation [CT&E], Security Test & Evaluation [ST&E]) in conjunction with normal system testing activities; support covert channel analysis, penetration testing, and operational testing."*

"7. Policy"

"b. Fundamental INFOSEC Policy"

"Section 6: Recommendations"

"Finally, for the long-term, the study team recommends that SPAWAR PD 51 pursue the analysis and application of certain classes of tools. These include tools that enable the ISSO/SA to monitor, probe, and analyze the security posture of an operational system (e.g., ISS, Icepick)."

Why is Ice-Pick's Use Acceptable

Ice-Pick, when properly used as an integral part of a network vulnerability protection program and is fully compliant with relevant individual privacy safeguards. It is not considered to be computer monitoring (in the legal sense) because it does not involve either real time wire interceptions, nor does it access stored communications. Since its use could present a 4th Amendment privacy concern, it is essential that the tester has the consent of those to be tested. Therefore, to protect both the tester and the test organization, formal authorization to test, signed by the appropriate authority must be obtained prior to testing, and all systems to be tested must have a security banner regarding expectation of privacy. The following basic model is recommended when a site is to be tested.

1. Identify the point of contact (usually the DAA) of the organization.
2. Get written permission from the point of contact to perform the vulnerability analysis
3. Notify system administrators on the target network (if appropriate)
4. Ensure that you properly select the approved specific target for testing
5. Do the vulnerability analysis (test)
6. Report all results to the organization's point of contact
7. Protect or destroy all vulnerability data collected still in your possession (as appropriate*)

*Ice-Pick has the ability to archive some test related information. If the tester is testing the site where he is employed and under direct supervision of the DAA, the data collected can be archived. If the tester is testing another organization's site, all vulnerability data should be delivered with no data archived.

The local site may also have an audit type monitoring tool requirement imposed on network test activities. This control function would then automatically provide a check on the testers activities as well as protecting the test authorizing organization from access liability. If such an audit tool is required, it is become the responsibility of the host organization to provide it to the Ice-Pick tester.

Inappropriate Use of Government Resources

What can happen to a tester if Ice-Pick is used in an unauthorized manner? Accessing, manipulating or otherwise using Government owned or leased equipment in an unauthorized manner, or on Government time, will be considered a misappropriation of public resources. Further, it is contrary to published Navy policy. If routine monitoring

by the IS Security organization reveals possible evidence of violation of criminal statutes, this evidence and any other related information, including identification information about the user, may be provided to law enforcement officials. If auditing or monitoring reveals violations of security regulations of unauthorized use, employees who are responsible may be subject to appropriate disciplinary action. The burden of responsibility rests directly on the user's shoulders should a potential legal issue develop later during an actual test.

Release Concerns

Predicting what would happen if a new vulnerability test tool was released without controls is difficult. Judging by what has transpired relative to the issuance of security advisories when similar programs were released, at the very least network attacks could noticeably increase. However, Ice-Pick's first line of defense is its internal program safeguards. The application is limited internally before distribution to pre-coded net masks.

The second line of defense relates to the trust safeguard. Unlike other available test tools, the Ice-Pick program is U.S. Government property and is strictly controlled for Official Government Use Only. Unauthorized use, distribution, reproduction, or possession may be grounds for criminal prosecution including imprisonment. As custodian of the Ice-Pick software, it is the user's responsibility to protect it.

How can user culpability be ensured? Through the use of training. Ice-Pick training covers applicable legal requirements as well as proper procedures and controls for tool application. Such mandatory training is also intended to reduce the possibility of accidental misuse as well as instill the importance of maintaining strict control of the software package.

The complete Ice-Pick package is a powerful security tool, useful for the system administrator to identify and fix potential vulnerabilities in Navy networks. If not protected, it could prove to be as useful to an unwanted perpetrator.

SECURITY THROUGH PROCESS MANAGEMENT

Jennifer L. Bayuk
Price Waterhouse, LLP
4 Headquarters Plaza North
Morristown, NJ 07962
jennifer_bayuk@notes.pw.com

Overview

This paper describes the security management process which must be in place to implement security controls. An effective security management process comprises six subprocesses: policy, awareness, access, monitoring, compliance, and strategy.

Security management relies on policy to dictate organizational standards with respect to security. Without policy, no person in the organization is responsible for securing information or is accountable for not having done so. A fundamental component of security management is a process for the production of security policy.

However, the resulting policy has value only if it is followed. A person who is not aware of an information security policy is not necessarily accountable for violating it. In the case of a system administrator configuring system security, ignorance of policy certainly provides an excuse to use personal judgment. Effective security management relies on an awareness process to provide accountability.

The policy process dictates *what* must be done to provide an acceptable level of assurance that systems are secure. The awareness process ensures that people know what must be done. To achieve assurance that policy is being followed uniformly throughout the organization, security management must also address *how* policy is to be realized. *How-to* solutions are effected via access and monitoring processes. Access and monitoring processes constitute the daily operational activities of security management. They provide guidelines on how to securely configure information systems and how to recognize a security incident.

Once a security incident has been recognized, a security management process requires methods to ensure that known security vulnerabilities are closed and open security issues are resolved. These methods are part of a compliance process. In addition, foresighted security management will include a strategy process to ensure that security management stays abreast of changes in the information technology environment which it seeks to secure.

Hence, an effective security management process comprises six subprocesses:

- Policy
- Awareness
- Access
- Monitoring
- Compliance
- Strategy

The Policy Process

A security policy is needed to establish a framework for the development of security procedures and practices. It also provides a vehicle with which to communicate roles and responsibilities with respect to securing information. A policy framework should specify the minimum security standards to be applied to all information systems, and more stringent standards for systems which contain highly sensitive or proprietary data. A security policy should address the following:

- Scope of the policy, including the facilities, systems, and personnel to which it applies
- Objectives of the security management process and descriptions of subprocesses
- Accountability and responsibility for subprocesses at all levels of the organization
- Minimum requirements for the secure configuration of all systems within the scope[†]
- Definition of violations and consequences of noncompliance
- A user statement of responsibility with respect to the information to which he or she is granted access

A security policy is a dynamic document. Its design should be flexible to allow frequent updates as technology and/or management changes require. Security policy development is not a project with a beginning and an end. A security policy coordinator should have responsibility for maintaining a policy team which is knowledgeable in both security techniques and the target information systems operating environment. The team leader must maintain open communications channels between the policy team, the management team who approves the policy, and those to which the policy applies. An example security policy process is depicted in Figure 1.

The Awareness Process

Though security personnel are arguably the best source of information concerning an individual's responsibilities with respect to information security, there are usually not enough of them to explain those responsibilities to everyone who falls within the scope of information security policy. It is enough that each department within scope designate an individual as a *security liaison*. Security personnel should create a security awareness program that may be implemented by department liaisons. This program needs to be flexible, comprehensive, clearly communicated, and understandable by department liaisons.

[†] Minimum security requirements for system configuration may contain standards which apply to only a subset of the facilities or organizations within the scope of the policy. For example, a security policy dedicated to a PC LAN environment will not make sense to implement on an IBM Mainframe. Designating specific sections or appendices to apply only to specific platforms enables the security policy as a whole to apply equally to all facilities, systems, and personnel within its scope.

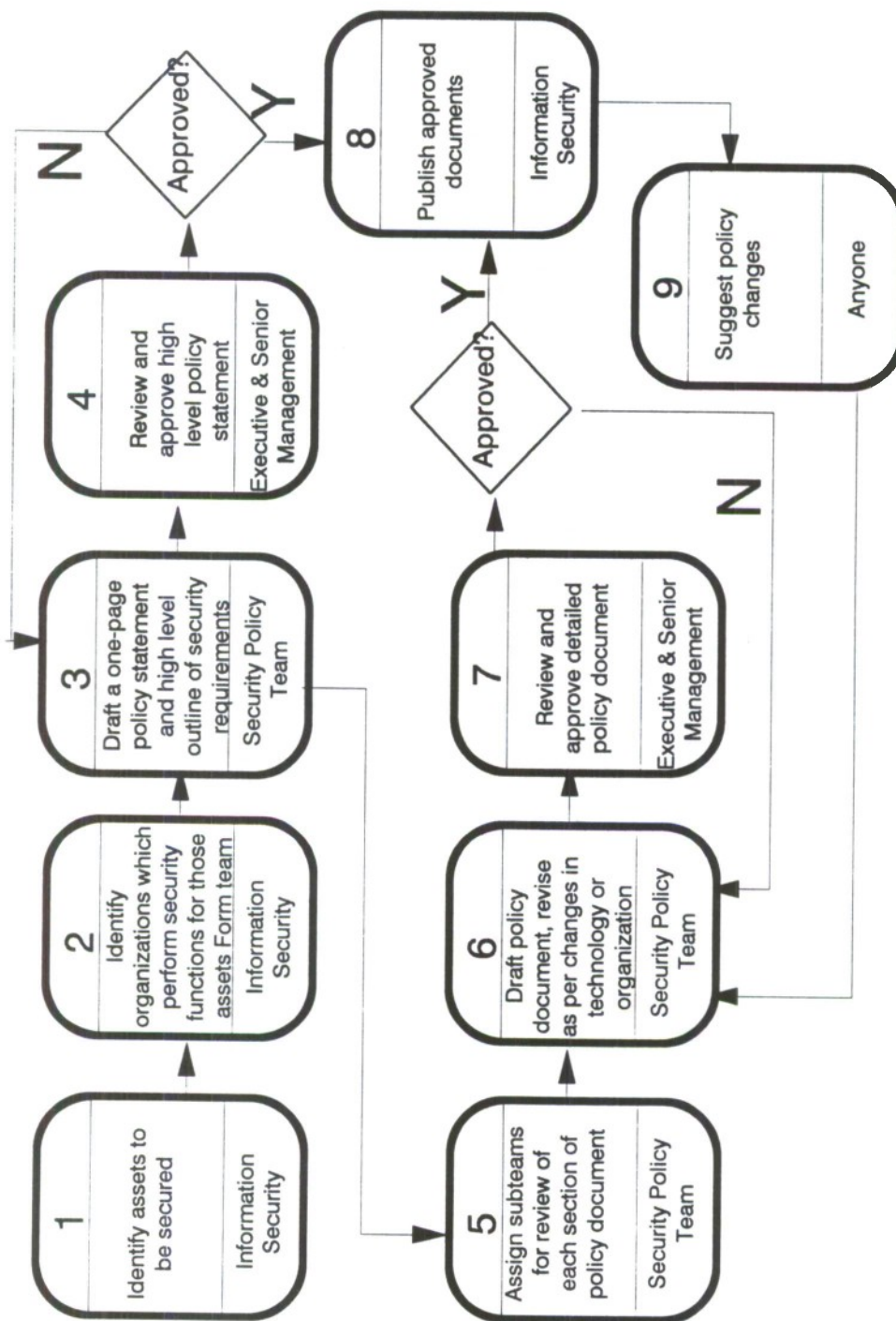


Figure 1: Example Policy Process

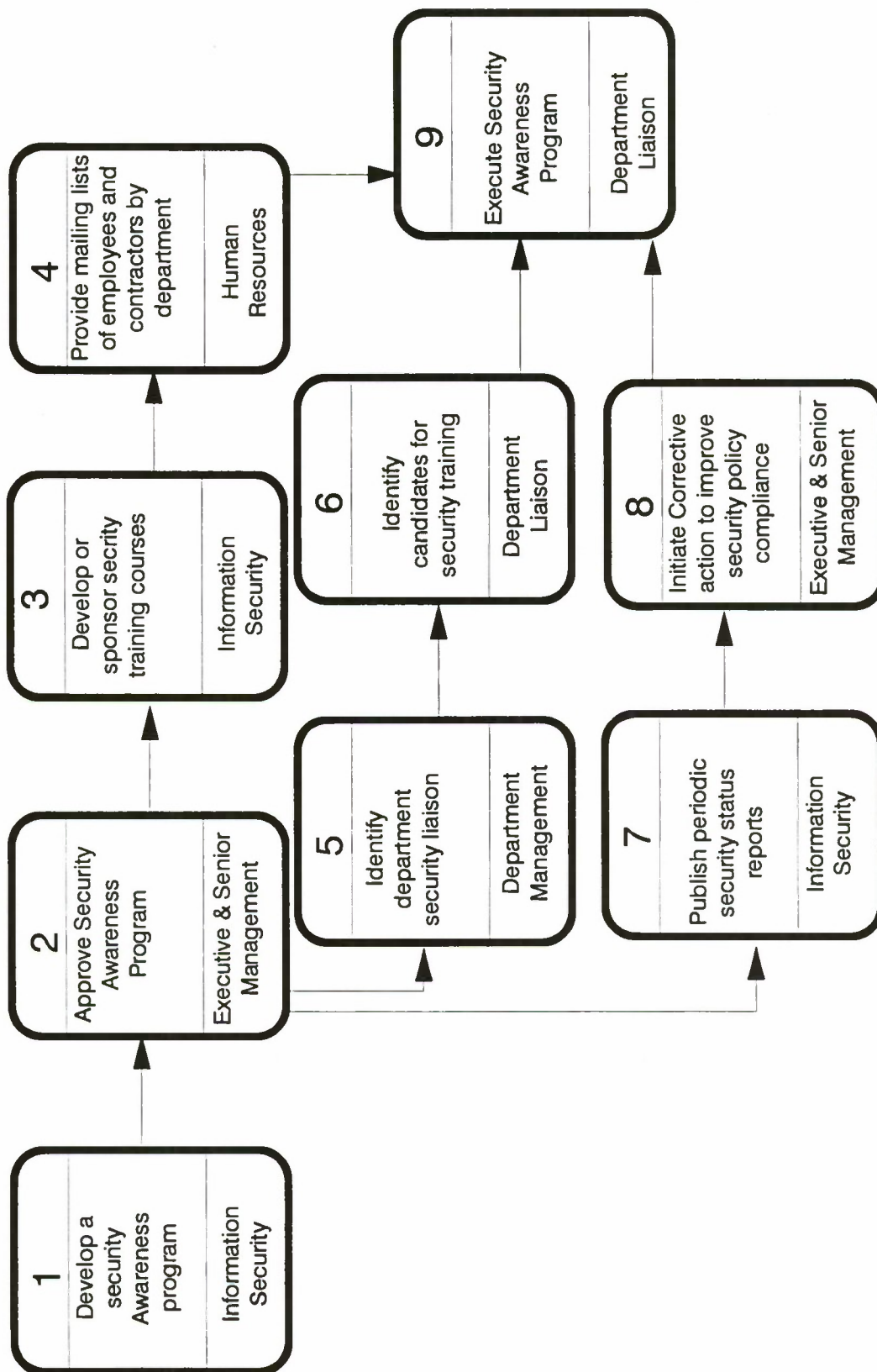


Figure 2: Example Awareness Process

The awareness program must clearly specify the actions required of employees and contractors and the seriousness of the actions that will be taken for non-compliance or violation of security policy. The awareness program should address the following key issues:

- Display high-level support
- Teach people how to obtain and comply with policy
- Point out the business risks in security policy violations
- Address the widest possible audience
- Allocate responsibility

An effective security awareness process will have executive and senior management play a formal role in improving security awareness by endorsing the security awareness program and by setting high priorities for security compliance. It will be integrated with personnel hiring and contracting practices to ensure completeness of coverage. The expected level of participation is evident in the example of a security awareness process depicted in Figure 2.

The Access Process

A security access process helps ensure that access decisions are made in a controlled manner, and that information concerning access is securely communicated between those that have a need to know. An access process should address:

- Identification of those who require access
- Authorization procedures for system access
- Automatic authentication of those identified and authorized for access
- Separation of duties between authorization and authentication
- Separation of access environments for distinct job responsibilities

Though security policy may dictate the details of how access should be administered, decisions concerning who should have access to production systems must rest solely with department managers responsible for the systems' smooth operation. In many organizations, the Information Security department facilitates the actual creation and maintenance of access, but it may be performed by anyone as long as it is done in accordance with policy and a separation of duties between authorization and authentication is maintained. An example access process is depicted in Figure 3.

The Monitoring Process

Security monitoring is required to detect both unauthorized system access and attempts at unauthorized system access. Left unmonitored, unauthorized access attempts may become unauthorized access. A security monitoring process includes three basic activities:

- configuring system security profiles and frequently reviewing system security logs
 - identifying the root cause of security alerts
- using the information derived from the first two activities to devise ever more meaningful system security profiles

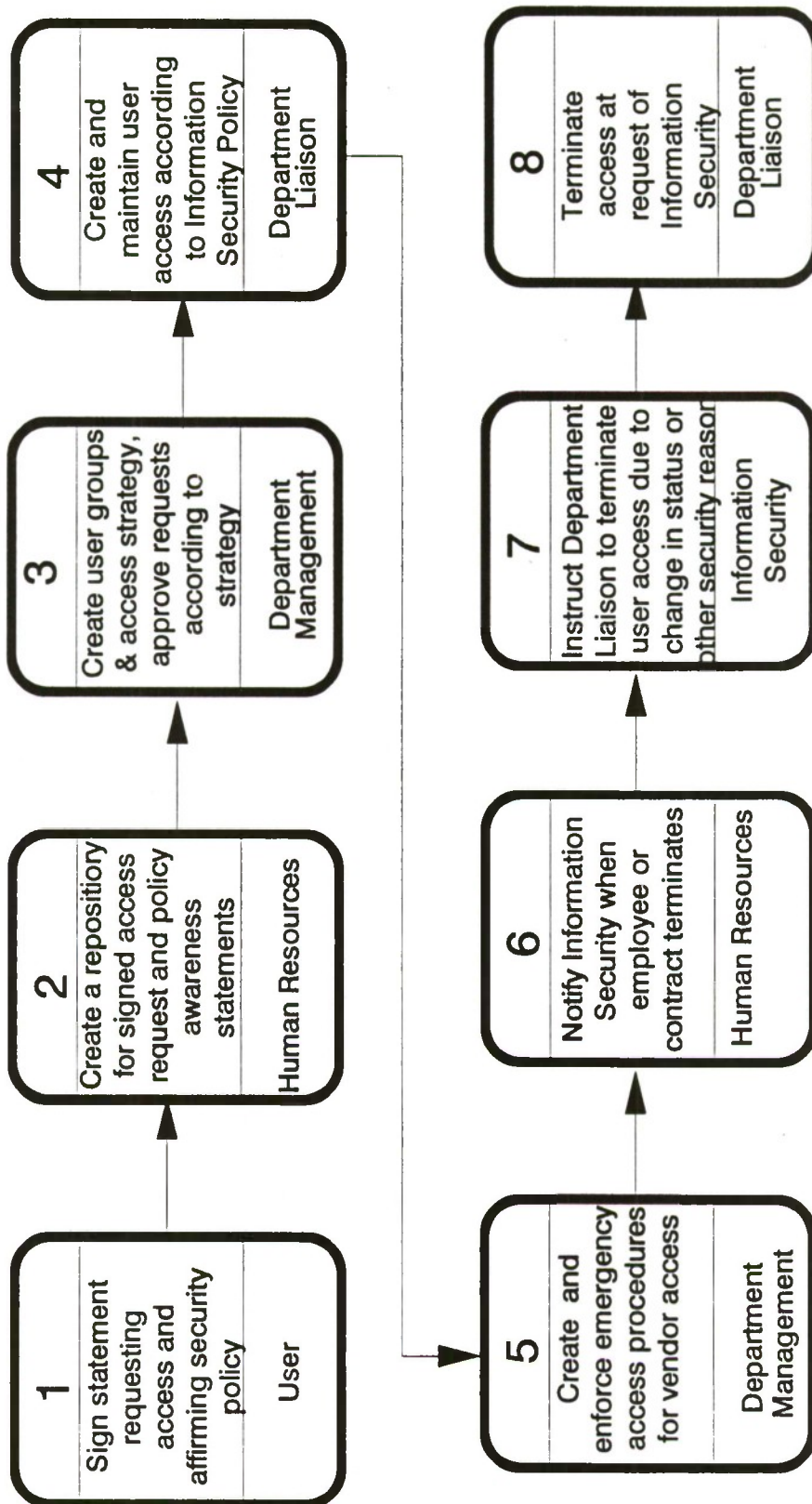


Figure 3: Example Access Process

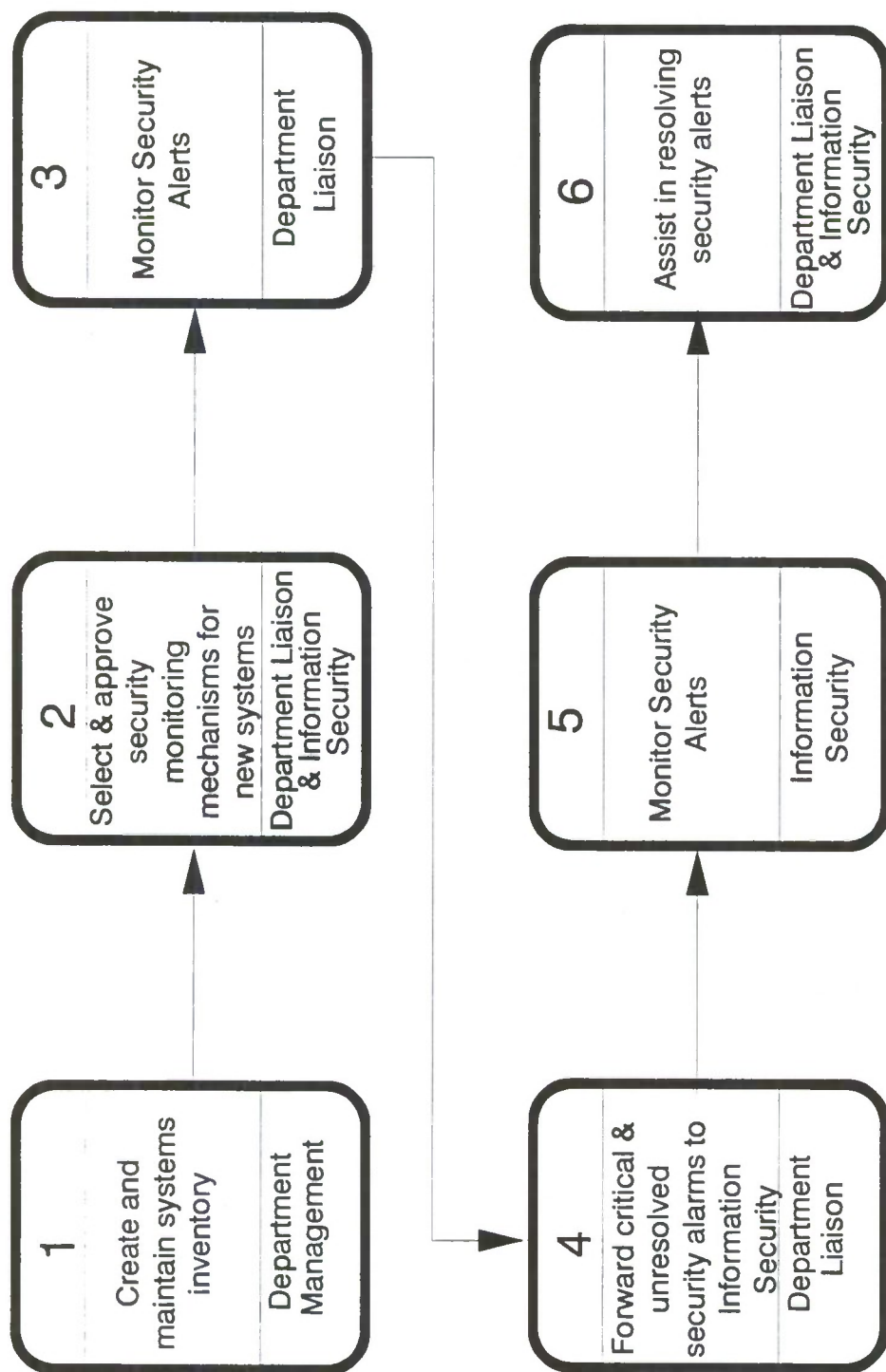


Figure 4: Example Monitoring Process

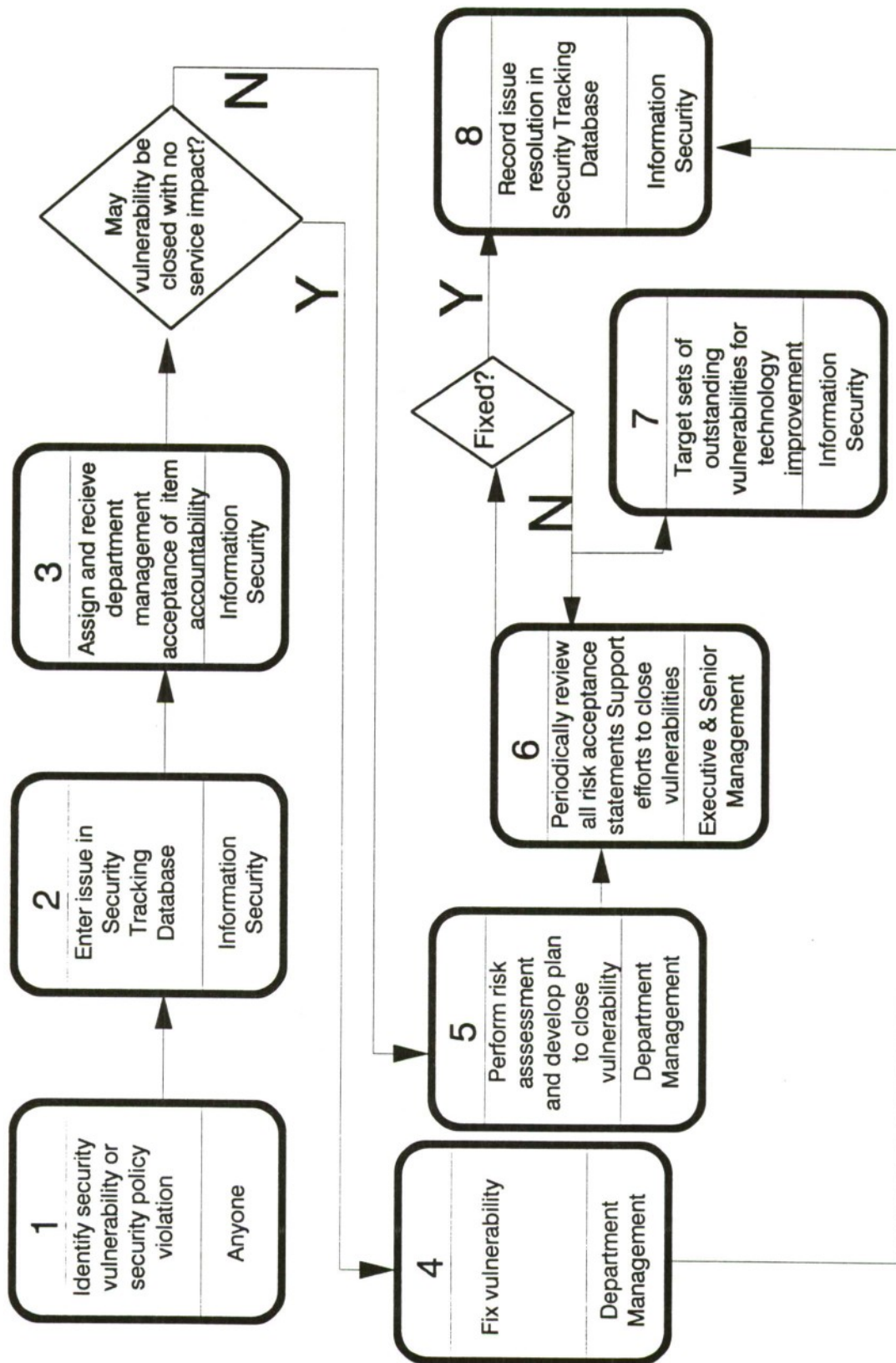


Figure 5: Example Compliance Process

Security monitoring is most cost-effective when merged with other monitoring processes such as performance or activity monitoring. However, where the root cause of a security alert cannot be determined or is determined to be a computer intrusion, system monitoring responsibilities should be shared with Information Security. Information Security should help ensure that all necessary operational and legal requirements for intrusion containment are met. Information Security should also track security alerts over time to determine if there are patterns. An example security monitoring process is depicted in Figure 4.

The Compliance Process

The extent to which there exists a formal compliance process is the extent to which security management efforts are effective in establishing a uniform level of security controls. Because compliance activities must be distributed among those who are responsible for the secure operation of information systems, departmental management must manage with reference to policies established by Information Security. However, there will be instances of non-compliance for many reasons, including:

- the technical architecture of a system does not support a required security function
- resources required to maintain compliance are unavailable
- a security incident reveals a security vulnerability which is not yet addressed by policy
- routine security audits or security reviews reveal previously unnoticed risks

In any case, the instance of noncompliance must be:

- reported to the Information Security
- assigned to appropriate management
- supported by a risk acceptance until resolved

The security compliance process must track all such security issues to ensure that steady progress is made toward their resolution. An example of a compliance process is depicted in Figure 5.

The Strategy Process

The security of information services is a reflection of the quality of information services. Developers of new products must recognize the strategic importance of integrating security mechanisms into the product itself. The security strategy process should aim to bring security expertise into long-range systems planning. The security strategy process may facilitate the integration of security into system design by:

- Developing risk assessment methods that quantify levels of operational risks in new products
- Contributing to business cases for including security mechanisms in architecture budgets
- Reviewing and testing new security features and products

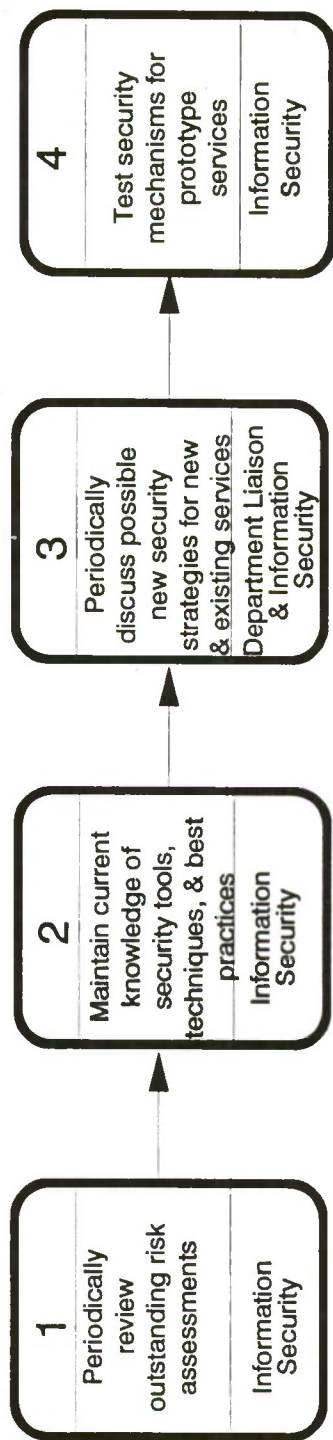


Figure 6: Example Strategy Process

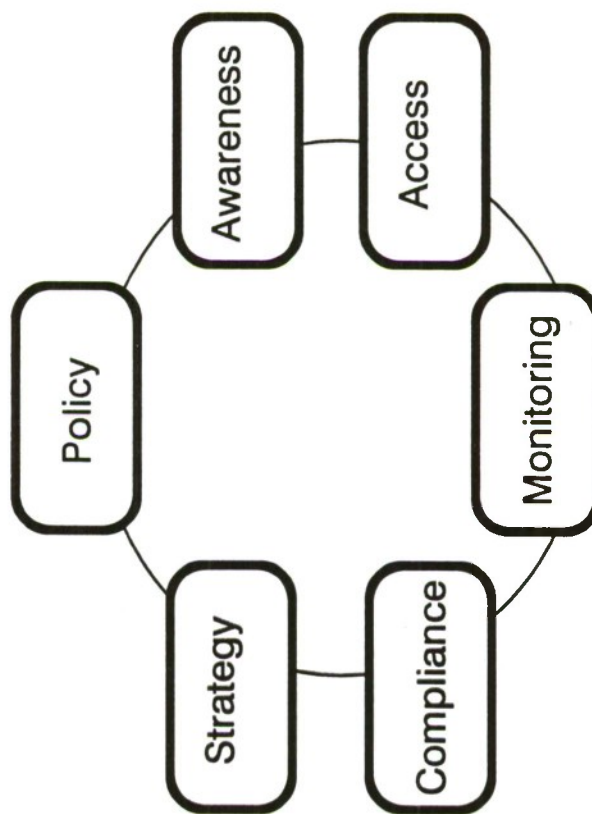


Figure 7: The Security Management Process

To facilitate the secure deployment of new and prototype services, the technical sophistication of the security department must equal that of the new service or prototype developer. Information Security must be an equal partner at every stage in the planning of services which require the use new technology. An example security strategy process is depicted in Figure 6.

Summary

This document describes the security management process which must be in place to implement security controls. The security management process includes six functions, each of which may be viewed as a distinct sub-process:

- Policy: to establish a framework for the development of organizational standards with respect to security
- Awareness: to educate those affected by security policy on their roles and responsibilities
- Access: to limit dissemination and modification of customer data and other sensitive information
- Monitoring: to detect policy violations and other security vulnerabilities
- Compliance: to track security issues and help ensure that resources facilitate the resolution of security issues
- Strategy: to meet the security challenges presented by new information technologies

Taken together, these six processes form one high-level security management process, displayed in Figure 7.

Malicious Data and Computer Security

W. Olin Sibert
InterTrust Technologies Corporation
460 Oakmead Parkway
Sunnyvale, CA 94086
osibert@intertrust.com

Abstract

Traditionally, computer security has focused on containing the effects of malicious users or malicious programs. However, as programs become more complex, an additional threat arises: malicious data. This threat arises because apparently benign programs can be made malicious, or subverted, by introduction of an attacker's data—data that is interpreted as instructions by the program to perform activities that the computer's operator would find undesirable. A variety of software features, some intentional and some unwitting, combine to create a software environment that is highly vulnerable to malicious data. This paper catalogs those features, discusses their effects, and examines potential countermeasures. In general, the outlook is depressing: as the economic incentives increase, these vulnerabilities are likely to be exploited more frequently; yet effective countermeasures are costly and complex.

1. Introduction

This paper addresses the increasing vulnerability of computer systems, particularly personal computers (PCs), to attacks based on *malicious data*: that is, attacks employing information that appears to represent input for an application program such as a word processor or spreadsheet, but that actually represents instructions that will be carried out by the computer without the knowledge or approval of the computer's operator. This vulnerability comes from two sources: program features that intentionally treat data as instructions and program flaws that allow data to act as instructions despite the program designer's intentions.

A system that has been subverted by such an attack is, in effect, under the control of a malicious program. Protection against such programs has been the focus of traditional computer security measures: file access control, user/supervisor state, etc. Such measures permit a program's activities to be contained to a limited set of computer resources for which the program's

operator is authorized. However, as computers (particularly PCs) are used more and more as extensions of their operators (i.e., as agents), the scope of authorization is greatly increased: a malicious program might, for example, cause a financial transaction using electronic commerce software that is *indistinguishable* by any automated means from a transaction the operator would have authorized—except that there was no such authorization. This increasing difficulty of identifying which computer activities are permissible and which are not increases the risk from *all* types of attacks.

The potential scope of malicious program activity in the PC environment is enormous. On one end of the spectrum are traditional "damage" attacks: virus propagation, destruction of data, compromise of other systems on a network. Another familiar attack involves disclosure: of passwords, of personal data, and so forth; but also of non-computer data such as credit card account numbers; see [1] and [2] for a detailed discussion of such a scenario and of how the disclosed data can be returned untraceably to the attacker. On the

other end of the spectrum are "agency" attacks, in which a computer is made to perform actions of which its operator is wholly unaware, such as electronic purchases, transfers of "digital cash," forged E-mail, and so on.

The two types of vulnerability from malicious data—intentional and unwitting—are quite different and require different approaches to remedy. The unwitting flaws can be fixed (although fixing them is rarely simple), but the intentional mechanisms represent a tension between a system designer's desire to provide features and a user's need for safety.

Furthermore, it is fundamentally difficult to distinguish between data and programs. Although many of the vulnerabilities discussed here rely on supplying actual machine instructions to be executed by hardware, others employ instructions that a program intentionally interprets (such as the PostScript language). Drawing a strict line between data and programs is not sufficient.

Section 1 of this paper introduces the concepts and discusses some potential effects. Section 2 catalogs a variety of intentional mechanisms that can be exploited using malicious data; section 3 describes unwitting mechanisms (i.e., flaws) with the same effect. Finally, section 4 discusses some solution approaches, of which disappointingly few seem to be effective.

2. Intentional Vulnerabilities

With the best of intentions, software developers are responsible for blurring the distinctions between programs and data. Most of the mechanisms cataloged in this section share a common characteristic: they provide a useful capability when used in a benign environment, but they were designed with little or no consideration as to how they might be employed by a hostile party (with the notable exception of Sun's Java language; also see section 4.2).

These mechanisms either permit arbitrary files to be modified, or allow arbitrary programs to be executed, or both. The fundamental property they share is an assumption that the operations that are performed should be performed just as if the user had entered

them directly at the keyboard: that is, they are executed within a "user environment" that is shared by all other activities the user performs. The difference is that these packages perform the operations without the user's explicit consent, and often without the user's knowledge. Although some of these features are undocumented, documentation is not the issue: it is simply unreasonable to expect a user to scour a 500-page manual looking for potential security risks before using a program.

Some of the risks posed by these mechanisms can be reduced or eliminated by isolation techniques or by requiring user confirmation. Such solutions, however, reduce the utility of the features and increase complexity for the user. Always requesting a confirmation is little better than never doing so: none but the most paranoid of users will think about it before answering "OK."

2.1. Examples

The following list identifies some of the intentional risks posed by common computer systems and applications:

- **PostScript file I/O primitives.** The PostScript language defines primitives for file I/O. These primitives can be used in a PostScript document to modify arbitrary files when the document is displayed; they extend PostScript's flexibility as a general-purpose programming language at relatively little cost in language complexity, but greatly increase its vulnerability to malicious data. Some PostScript interpreters (e.g., Display PostScript, GhostScript [3]) can disable these primitives (and other non-imaging functions), but doing so is not always simple and can also be seen as inhibiting desired functions.
- **Pathnames in archives.** Arbitrary file pathnames can be stored in common archive formats (i.e., using a maliciously modified tar or PKZIP program) so that unpacking the archive potentially, can overwrite arbitrary files.
- **Application startup macros.** Much "desktop productivity" software, such as Lotus 1-2-3 and Microsoft Word, provides the ability to run a macro when a data file or document is first opened.

Often, the macro languages include file I/O primitives or even permit execution of arbitrary commands, thus enabling such a “document” to perform arbitrary actions just because it is viewed. Similar features are also present in older applications (for example, the UNIX `troff` document processor has a request for executing a command line while processing a document).

This problem has been understood theoretically for a long time [4][5], but its first exploitations “in the wild” occurred only recently: the Microsoft Word “Concept” Virus [6]. When opened, any document containing this virus modifies the user’s environment so that all future Microsoft Word documents will carry it. It is possible to prevent the automatic execution of start-up macros by pressing the SHIFT key while selecting a document, or by disabling the feature globally. Despite publicity, however, it seems unlikely that most users will know how to perform countermeasures, or that they will always remember to do so.

- **Automatic system actions.** A variation on this theme is the feature in certain operating systems (such as “Autoplay” in Microsoft Windows 95), that automatically invokes a program stored on a CD-ROM or other media when the media is inserted into a computer system. This feature may be convenient for media that one trusts (perhaps with a digital signature to provide proof of origin), but it represents a major risk for arbitrary disks. The “read-only” nature of the media is no protection: with the advent of inexpensive CD-ROM writers, writable CD-ROM has become widely used for data interchange.
- **Executable Mail Attachments.** Many modern mail readers provide the ability to attach arbitrary objects to a message—including executable programs. The obvious thing to do with such an attachment is to select it, as one might select a document attachment in order to view it. Although this act is clearly a discretionary one by the user, it is also a very natural one, and the system gives no hint that it might be more dangerous than, say, viewing an image file.

An early version of this attack was the “CHRISTMA EXEC” virus that propagated on IBM’s internal network in 1987 [5]. The mail system did not facilitate this attack: rather, explicit

action was required to write out the file and run it, but even so, users almost invariably followed the instructions and did so without suspicion.

- **Executable Web content.** Some Web browsers (e.g., Netscape Navigator, Sun’s HotJava, Microsoft’s Internet Explorer) offer the capability of downloading and executing parts of a web page locally. In some cases (e.g., Java programs) the local execution is strongly constrained for security reasons and relatively safe; in others (e.g., Explorer and downloaded OLE controls) there are no restrictions on the code being executed.

3. Unwitting Vulnerabilities

The previous section dealt with purposeful software features that provide an opportunity to introduce malicious programs. On the one hand, it is unfortunate that those features were designed with little attention to risk; on the other hand, it is good that they can be identified, for it is possible to imagine countermeasures that would contain them.

There is another class of attacks that does not have those properties: attacks based on program flaws or inadequate design. Here, the designers did not intentionally create a problem; rather, by failing to provide sufficiently robust software, they unintentionally enabled the problem to occur.

Such unintended risks depend on the same basic properties as the intentional ones: programs run in a user environment that is shared by other programs. To date, the exploitations of these risks have involved primarily multiuser systems, where the environment being attacked is privileged. However, privilege is not necessary for these attacks to be useful; they can introduce malicious software into the environment of an unprivileged user just as effectively.

3.1. Examples

A few examples of these attacks include:

- **The Morris Worm fingerd Attack.** As reported in [7], the “Morris Worm” delivered an executable program over a network connection to

the `fingerd` program and, by overflowing an internal buffer where the request was stored, caused it to be executed.

This attack, part of the incident that brought the Internet to a halt in 1988, relied on the presence of a fixed-size buffer inside the `fingerd` program. The request received from the network was read into the buffer without a check on its length. Because the request was larger than the buffer, it would overwrite other data, including the return address stored in the stack frame. By changing the return address to designate a location within the request string, the attack forced a transfer of control to the attacker's supplied program stored in the request string. When executed, this small program established a run-time environment and carried out the rest of the attack. This attack was very sensitive to initial conditions: it was developed only for one widely-used operating system, and it depended on the stack frame layout in the `fingerd` program, the containing process environment, and so forth. However, given the source code to the `fingerd` program and a laboratory system on which to experiment, it appears that the attack was engineered with only a few days of effort.

This attack on the `fingerd` program was the first widely demonstrated example of forcing an application program with no intentional programmability features to execute machine instructions supplied by an attacker. It required only a modest engineering effort to create, and it was wildly successful. It breached internal security in multiuser computer systems, which is not normally an issue in personal computers, but it pointed the way for similar attacks in different environments.

- **The Netscape Navigator attack.** In late 1995, a flurry of security problems with cryptography and random number generation in Netscape's Navigator program was reported in the mainstream press. Shortly afterward, some members of the "Cypherpunks" group discovered a buffer overflow flaw in the then-current version of Navigator. This attack is notable because it is directed at a personal computer program, where the objective is not to breach multiuser security but to cause a personal computer to act under control of malicious software.

Using techniques similar to those used in attacking the `fingerd` program, an **over-length** host name in the HTML source of a **Web page can be made to**

overflow an internal buffer and cause an attacker's program to be executed. Although Navigator's parsing of the HTML language itself turned out to be fairly robust, the routine that converted a host name to an Internet address had a fixed-size buffer that could be overwritten by an oversize fabricated host name, and this led to the ability to cause Navigator to branch to an arbitrary execution address.

- **Overflow `syslog` buffer.** Like `fingerd`, the `syslog` program used for system logging in UNIX systems was found in 1995 to be vulnerable to buffer overflow[8]. The attack technique and the objective (run a program in a privileged process—in this case, `sendmail`) are essentially equivalent to the `fingerd` attack. Although much attention has been paid to eliminating such vulnerabilities in the intervening seven years, the continual emergence of examples suggests that it is very difficult to eliminate the problem systematically.

3.2. Scope of Vulnerability

These examples represent the tip of the iceberg. What sort of programs are vulnerable to such attacks? Any program that misbehaves when given bad input data is a potential victim. If it crashes or dumps core when given bad input, it can probably be made to misbehave in a predictable manner, too. If a program's internal data structures can be damaged by invalid input, this often indicates that its control flow can be affected as well—potentially leading to the ability to execute caller-supplied instructions.

Indeed, software developers typically make no claims that any application programs are bulletproof when faced with invalid input data, because such misbehavior is seen only as an inconvenience to users—after all, "garbage in, garbage out." The risk that it would serve as a way to introduce malicious software into the user environment is rarely, if ever, considered.

Examples of such program misbehaviors include:

- The UNIX utility `uncompress` often dumps core when processing invalid compressed input data. The `gunzip` or `PKUNZIP` decompression utilities may have similar problems.

- Programs that process complex data formats, such as MPEG streams, Rich Text Format, or PostScript may produce wildly incorrect output or misbehave when given malformed input data, indicating that their internal data structures have been damaged.
- Application software such as Microsoft Word or Lotus 1-2-3 often fail catastrophically when given a damaged input file, again likely indicating damage to internal data structures.
- File import software (e.g., Microsoft Word reading WordPerfect documents) often is even more fragile than the software processing an application's native format, making it more likely to harbor vulnerabilities. This fragility may occur because these parts of the software are less thoroughly tested, because they are written by third parties, or perhaps because the formats being converted are themselves undocumented.

Although none of these program behaviors is known to the author to have been exploited, the possibility clearly is present, and further investigation is warranted.

The basic problem is that increasingly complex and ill-defined data semantics are difficult to process, so it is no surprise that application software fails when presented with bogus input data. Software that responds correctly to all incorrect input is far harder to create than software that simply responds correctly to correct input.

Application software development contrasts with the design philosophy of network protocols, where a basic assumption is that all possible bit sequences will be encountered, so all must be handled reasonably. It is partly for this reason that implementations of complex network protocols often have a long development period before they are truly robust.

3.3. Exploitation Techniques

The known exploitations for invalid input data are known primarily because they were used to breach system integrity in multiuser systems. These attacks are more difficult to construct than those that exploit known software features. They require constructing executable programs “by hand,” tailored to run in a

largely unknown environment. Although doing so is awkward, it is by no means beyond the abilities of a moderately sophisticated attacker.

The most fruitful exploitation technique seems to be buffer overflow: provide more data than a program expects, so that it will overwrite internal storage for program variables or addresses, and the program will misbehave in a deterministic—and possibly controllable—manner. Another technique involves providing data with out-of-range values. Such inputs can cause calculated branches to go to unintended destinations, or can cause values to be stored outside of array bounds. All these vulnerabilities offer the potential to cause a transfer to the attacker's supplied executable code, from which point the attacker can do anything that the attacked program can do.

It is important to note that malicious data representing machine instructions does not require arbitrary binary values. For example, the Intel 80x86 opcode set and the MS-DOS executable file format permit a valid executable program to be constructed entirely from printable ASCII characters. Such a program can perform arbitrary actions when executed, yet it requires no special transfer mechanism—it can be delivered as ordinary unformatted E-mail. The first known example of such a program¹[9] contains a small executable header that decodes the rest of the program text—transferred in UNIX uuencode format—into a memory buffer, then transfers to it.

Of course, a successful exploitation is quite difficult. It is necessary first to understand how the program is misbehaving, then to determine what input data will create predictable misbehavior, then to craft input data that contains an appropriate attacking program. The analysis stages require an understanding of the software that comes most readily from source code, but as most personal computer applications are not distributed in source code form, techniques such as disassembly and emulation are required. Experimentation plays a critical role, also.

1. For example, when saved as a text file, the following five lines of text (Copyright © 1994 by A. Padgett Peterson) form an executable MS-DOS program that prints a short text message:

```
XP[0PPD]5'P(f#(f(75!QP_u!2$=po)l=!!rZF*S* $=0%GF%!!%PP$P$P$-
%gmZ$xr16lW$rm6mWlV16m=ldmAlv%fmvmB$Vm6lW$Vm6mWl6m6m=ld%y1VmqlJmq
lRmq1NmqlB1W16m6l/m'1/m3mWl7m7mrm4mq1:lXl7m7mAl2lY1l6m6lZl6m2mPm
mPl%0%[$'SU%^$/$%bl%Y$X%[%$Z%Yl%$q%b%$a%'^l%W$%^$'\%`l%5p%`
$as%a%`l%$%[%`%Yl%$p%b%$/$%b%Y$|l%$[%`%W%Yl%$b%$Xl%$%z%$|l$pp
```

4. Solutions

Solving the problems posed by unsafe or malicious data requires fundamentally different techniques from traditional computer security approaches, because the objective is different. Traditional approaches focus on isolation and protection of resources: that is, on preventing activity whose nature is known in advance. Protection from malicious data, on the other hand, requires distinguishing among program activities that are in accord with the operator's intent and those that the operator would not want to occur. This problem—of divining the operator's intent—seems unlikely to be solved.

Addressing the malicious data problem seems instead to require a return to fundamentals:

- **Avoid building unsafe features into computer programs.** This would reduce the incidence of "intentional" problems.
- **Use programming techniques and languages that encourage construction of robust programs.** This would reduce the frequency and severity of "unwitting" vulnerabilities.

Aside from these techniques—which would represent a fundamental change in commercial software development—there are relatively few external, system-level techniques that offer much hope for improvement. The problem of safe execution of mutually suspicious programs remains a difficult problem in computer system design [10]. Even if such solutions were readily available, it is unclear whether users could be expected to exercise the necessary discipline to protect themselves. After all, it is not unreasonable to expect that computer systems, like other complex appliances, should be safe to use without detailed understanding of their internal operations.

4.1. External Solutions

This section briefly discusses some of the solution techniques that can be applied externally to contain or reduce the effects of malicious data:

- **System isolation.** A computer system that is not connected to a network and used for only one purpose is unlikely to be vulnerable to malicious data,

and even if attacked, would not be able to do much damage. This approach is, by default, what has protected most personal computers—but increasingly these computers are networked and used for many activities.

- **Virtual machine environments.** Suspect or untested software can be run under control of a virtual machine monitor; this approach in effect is the same as running many isolated systems. As long as the virtual machines remain isolated, this technique contains the problem effectively, but as soon as data is transferred among them, they become vulnerable. Maintaining the necessary isolation requires a generally infeasible degree of discipline on part of the operator. It is not reasonable to expect personal computer operators to maintain a constant state of suspicion.
- **Automated filters.** Known examples of malicious data can be detected and filtered out. For example, Secure Computing Corporation's Sidewinder product can analyze all traffic coming across a network firewall and reject patterns that it recognizes as malicious (such as virus-infected executables or malformed HTML documents). Similarly, some virus detection products are now capable of detecting the known examples of the Microsoft Word virus described in section 2.1.
- **Capability-based operating systems.** Capability systems were a major focus of operating system research in the 1970s [11]. In principle, such systems can safely contain the effects of malicious data more effectively than virtual machine monitors because they exercise control over resources at a finer grain. However, capability systems have the same drawback of requiring considerable discipline to use effectively and also require special hardware and/or programming techniques to use effectively. Although a few capability-based systems were introduced in the 1980s (from companies such as BiiN, Intel [12], and Key Logic [13][14]), these were not commercially successful, and they are no longer actively marketed.
- **Dynamic monitoring.** The virus protection field deals with some of the problems that can be caused by malicious data. One of the techniques developed for virus protection is dynamic monitoring of program activity: pattern matching of program opera-

tions against acceptable types of operations [15] (e.g., files to which a program is expected to write, as opposed to those to which it should not). A user can be presented with the opportunity to permit or deny such actions.

- **Digitally signed executables.** Public-key cryptography can be used to sign application software and certify it as “safe” as judged by some certifier—where one of the “safety” properties would be that the application cannot be corrupted by malicious data. This technique has been proposed as a way of marking executable Web content as safe to use. Unfortunately, it simply moves the burden of assurance to a certifier without making the analysis any more tractable; it also places an unreasonable burden on users, who must decide which certifiers are trustworthy. Because even major mass market application software appears susceptible to malicious data attacks, it is not clear what value this type of certification technique could add.

4.2. Internal Solutions

In the long term, internal solutions seem to offer more hope for addressing these problems:

- **Safe application design.** Defense against intentional mechanisms that permit malicious data to be introduced requires that application designers pay more attention to system safety. That is, they should avoid features that introduce unconstrained programmability into an application.
- **Safer languages.** The most important defense against malicious data is programs that are more resistant to it. An important part of this resistance involves use of languages and environments that are themselves robust, with bounds checking, pointer validation, memory management, and so forth. The Java language [16] is one such; others (e.g., Ada and Python [17]) also have extensive robustness features.

The Java language is particularly interesting because of its program validation mechanism and its utility for enforcing type safety rules to contain features that could introduce intentional vulnerabilities. Unfortunately, current versions of Java do not live up to the promise of safe execution. Although some of the problems reported in [18] and detailed

in [19] result from simple implementation problems related to specific execution environments, two design flaws have been reported that breach the type safety of the language itself. The lack of a formal basis for Java’s claimed type safety and security properties is troubling.

- **Non-von Neumann computer architectures.** The principal mechanism for unintentional malicious data flaws is the ability to execute data: an attacker supplies malicious instructions as data and causes a branch to them. If instructions are clearly distinguished from data, the attack is much harder. Unfortunately, the prevalent use of interpreters, sometimes with multiple levels of interpretation, makes this approach unworkable on a hardware level.
- **Sheer complexity of applications.** One reason that these attacks have not been more widely perpetrated is that they are *difficult*, because much application software is not available in source code form and is extremely complex. An attacker must understand a great deal about a program’s internal operation to be able to fabricate malicious data that will cause predictable types of misbehavior. Although not a defense one would like to rely on, it has been reasonably effective.

5. Conclusions

The general outlook for malicious data as a computer security problem is unclear. The potential vulnerabilities are legion, but exploitation poses great practical difficulties. Unfortunately, defense also poses great difficulties, and as the economic incentive for creating malicious software increases, it seems likely that attackers will attempt to exploit these vulnerabilities.

The most effective technical solutions appear to require pervasive change in the way that computer software is built. The near-term alternatives all involve giving up many of the “general-purpose tool” properties that make personal computers so effective in the first place.

6. References

- [1] Garfinkel, Simson, "Program shows ease of stealing credit card information," San Jose Mercury News, 29 January 1996
- [2] Sibert, Olin, "Risks (and lack thereof) of typing credit card numbers" Risks-Forum Digest, volume 17, issue 69, 7 February 1996, available by anonymous FTP from ftp.sri.com in /risks/17/risks-17.69
- [3] Computer Emergency Response Team, *CERT Advisory CA-95:10*, 31 August 1995, available by anonymous FTP from info.cert.org in /pub/cert_advisories/CA-95:10.ghostscript
- [4] Hoffmann, Lance J., *Rogue Programs: Viruses, Worms, and Trojan Horses*, Van Nostrand Reinhold, 1990
- [5] Ferbrache, David, *A Pathology of Computer Viruses*, Springer Verlag, 1992
- [6] Computer Incident Advisory Capability United States (Department of Energy), *CIAC Alert G-10a: Winword Macro Viruses*, available from <http://ciac.llnl.gov/ciac/bulletins/g-10a.shtml>
- [7] Eichin, Mark W., and Rochlis, Jon A., "With Microscope and Tweezers: An Analysis of the Internet Virus of November 1988," in *Proceedings, 1989 IEEE Computer Society Symposium on Security and Privacy* page 326-343, 1-3 May 1989, Oakland, California
- [8] Computer Emergency Response Team, *CERT Advisory CA-95:10*, 19 October 1995, available by anonymous FTP from info.cert.org in /pub/cert_advisories/CA-95:13.syslog.vul
- [9] Peterson, A. Padgett, personal communication, 15 February 1996. Mr. Peterson reports, "I described it in an internal Martin Marietta memo on security threats presented in 1988 as 'theoretically possible' but did not construct a working prototype [until 1994]."
- [10] Rotenberg, Leo J., *Making Computers Keep Secrets*, Ph.D. Thesis, Massachusetts Institute of Technology, 1973, published as MIT Project MAC Technical Report TR-115, February 1974
- [11] Levy, H. M., *Capability-based Computer Systems*, Digital Press, Maynard, Massachusetts, 1984
- [12] Organick, Elliott I., *A Programmer's View of the Intel 432 System*, McGraw-Hill, New York, 1985
- [13] Hardy, Norman, "KeyKOS Architecture," *ACM Operating Systems Review*, Volume 19, Number 4, October 1985
- [14] Rajunas, Susan, et al., "Security in KeyKOS," in *Proceedings, 1986 IEEE Computer Society Symposium on Security and Privacy*, 7-9 April 1986, Oakland, California
- [15] Pozzo, Maria, and Gray, Terrence, "Managing Exposure to Potentially Malicious Programs," in *Proceedings of 1986 National Computer Security Conference*, 15-18 September 1986, National Bureau of Standards, Gaithersburg, Maryland pages 75-80
- [16] Sun Microsystems, Inc., *Java Language Specification*, available as <http://www.javasoft.com/JDK-beta-2/psfiles/javaspec.ps>
- [17] van Rossum, Guido, *Python Reference Manual*, Dept. AA, CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, available as <http://www.python.org/doc/ref/ref.html>
- [18] Computer Emergency Response Team, *CERT Advisory CA-96:07*, 29 March 1996, available by anonymous FTP from info.cert.org in pub/cert_advisories/CA-96:07.java_bytecode_verifier
- [19] Dean, Drew; Felten, Edward; and Wallach, Dan, "Java Security: From HotJava to Netscape and Beyond," in *Proceedings, 1996 IEEE Computer Society Symposium on Security and Privacy*, 6-8 May 1996, Oakland, California

Security Issues for Telecommuting

Lisa J. Carnahan and Barbara Guttman
Information Technology Laboratory
National Institute of Standards and Technology

Abstract

Telecommuting affords many in the workforce options about where and when they can work. Many organizations are promoting telecommuting to allow their employees to work from home, while on travel, at a client site, or in a telecommuting center. While the benefits to telecommuting are obvious, new risks to the organization are introduced. This paper will highlight security issues related to telecommuting and propose solutions that may help organizations better manage the telecommuting environment.

The Risk of Telecommuting

Telecommuting is one of the popular buzz words for management in the '90s. It is becoming accepted as the way to do business. However, opening up corporate¹ systems to dial-in and other forms of access presents three significant security risks.

The first risk is that intruders will be able to access corporate systems without having to be on site. Hackers armed with war dialers, electronic eavesdroppers at conference sites, or shoulder surfers watching employees enter IDs and passwords are all very real threats in today's environment. In addition to intruders whose goal may be mischief, hacking is attractive to people trying to steal or misuse corporate information. Electronic access to records is often more anonymous than trying to bribe employees or gain physical access.

A second risk of telecommuting, closely related to the first, is that corporate information can be read, and potentially modified, while it is in transit.

Telecommuting also presents organizations with more pedestrian risks. These include the risk of losing corporate information and resources when they are outside the protective shell of the organization.

What is *telecommuting*? It is the use of telecommunications to create an "office" away from the established (physical) office. The telecommuting office could be in an employee's house, a hotel room or conference center, any site an employee travels to, or a telecommuting center. The telecommuter's office may or may not have the full computer functionality of the

¹ Corporate is used to mean the belonging of any organization, including public sector agencies, private sector business, academic institutions, or other types of organizations.

established office. For example, an employee on travel may read email. On the other side of the spectrum, an employee's house may be equipped with ISDN and the employee may have full computer capability at high speeds.

Security Issues for Protecting Internal Systems

In planning for the security of telecommuting, the first step is to examine what type of access is needed. What systems and data do employees need? What is the sensitivity of these systems and data? Do they need system administrator privileges? Do they need to share files with other employees? Is the data confidential?

From a security perspective, the critical determinations are:

- What would happen if an intruder gained the same access as the employee?
- What would happen if an intruder were able to use the employee's account, but gain more access than authorized for that user?

If the answer to either of these questions is "uh-oh," then security is important.

A. Firewalls/Secure Gateways

A secure gateway, often called a firewall, blocks or filters access between two networks, often between a private network and a larger more public networks such as the Internet or public switched network (i.e., the phone system). For telecommuting, the trick is to decide what to make available to telecommuting employees using public networks, what degree to ensure that only authorized users can get to the internal network, and how to ensure that the secure gateway works properly.

If possible, it can be more secure to put all the resources needed by telecommuting employees outside of a secure gateway. However, this is only possible if employees do not need access to corporate databases. For example, employees may only need to send reports in or access public databases, such as product/sales information or government forms.

However, most telecommuting employees will need more access. For traveling employees, this may be limited to needing email. There are many firewall implementations that use a email proxy to allow access to the files on a protected system without having to directly access that system.

Once again, many telecommuting employees will need more access. They need access to internal resources. The employees may need to use a variety of resources such as LAN applications, mainframe applications, run client software, use TCP/IP services.

A secure gateway, or series of gateways, can be used to divide internal resources based on access need of telecommuters. For example, computers with high-risk organizational data

(such as proprietary business plans) may be separated by router from systems with a lower level of risk. A series of routers can be used to further restrict access to the highest-risk systems.

For some situations, current firewall technology can be used to give virtual access by using proxies. In addition, current firewall can use IP filtering to permit access to only certain types of resources.

However, for many organizations, the primary security function of the secure gateway is to provide robust authentication of users.

Secure gateways may also provide additional auditing and session monitoring. The gateway can perform an intrusion detection function. For example, the secure gateway could monitor a session for keystrokes which may indicate someone trying to exceed access (e.g., ^C, ^Z).

B. Robust Authentication

For most organizations, robust authentication should be required if access is given to internal systems. However, many organization should require robust authentication even for email if it is relied to discuss business decisions (i.e., if the organization would care if someone else read your email).

Robust authentication can increase security in two significant ways: 1) It can require the user to possess a token in addition to a password or PIN and 2) it can provide one-time passwords. Tokens when used with PINs provide significantly more security than passwords. For a hacker or other would-be impersonator to pretend to be someone else, the impersonator must have both a valid token *and* the corresponding PIN. This is much more difficult than obtaining a valid password and user ID combination (especially since most user IDs are common knowledge).

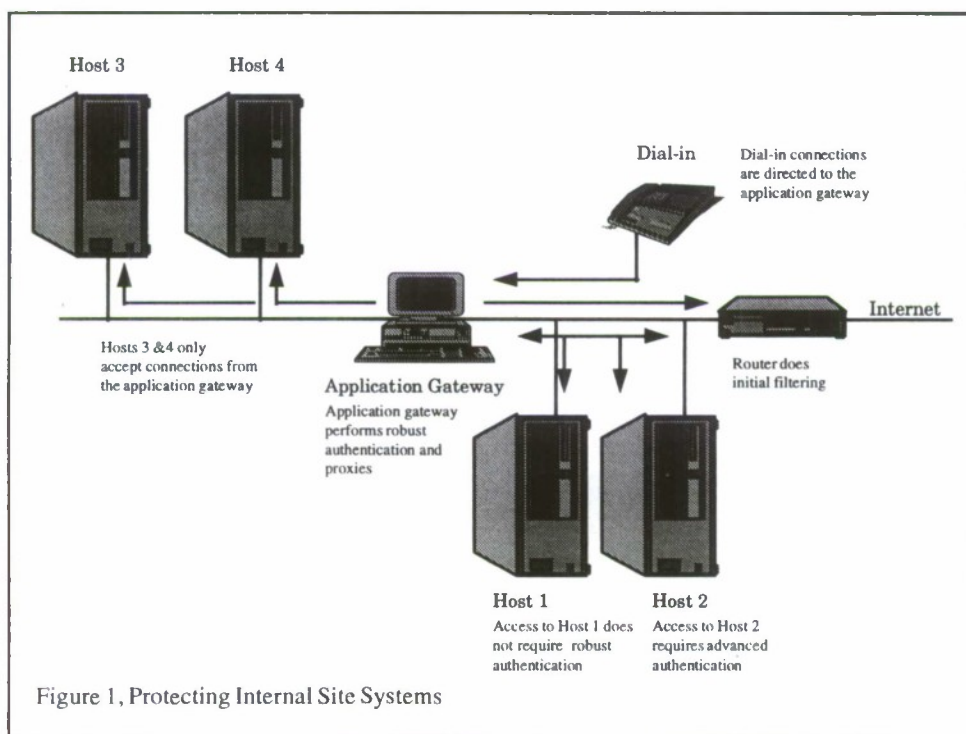
Robust authentication can also create one-time passwords. Electronic monitoring (eavesdropping or sniffing) or observing a user type in a password is not a threat with one-time passwords because each time a user is authenticated to the computer, a different "password" is used. (A hacker could learn the one-time password through electronic monitoring, but it would be of no value.)

Most commercial robust authentication systems use smart tokens. The user provides a PIN which unlocks the token and then uses the token to create a one-time password. However, it is possible to use software-only one-time password schemes. (Tokens which do not provide for one-time passwords, such as ATM cards, are less common for telecommuting because they require hardware at the remote site and, without physical security, are vulnerable to electronic monitoring.)

Telecommuting employees who directly access internal systems should be robustly authenticated and should be routed to specific computer systems. The combination of routing and robust authentication can greatly increase security and reduce the costs associated with

robust authentication by limiting it to employees with the greatest access.

The following figure diagrams an example of an agency with multiple access points for telecommuting that segregates telecommuters into three risk-based areas. Access to Host 1 is granted based on simple password-based authentication. Host 1 contains read only applications. There is no confidential data on Host 1. Access to Host 2 is granted based on robust authentication, but is outside the firewall. The rationale for creating Host 2 was to be able to support applications that the firewall cannot protect against (e.g., no proxy is available). Access to internal systems (Host 3, Host 4 and the LAN) requires robust authentication. The firewall uses proxies to mediate between the external network (including both Internet and dial-in connectivity) and the internal network.



Three caveats need to be made:

1. Any additional logins (to Host 3 or Host 4, for example) are in the clear. Anyone eavesdropping on the connection can gain a valid ID and password to Host 3 or Host 4. With proper configuration management (i.e., no modem connections inside the firewall), these systems will not be directly accessible from the outside, and, therefore, the ID and password will not be usable.
2. Too much or too complicated segregation may prevent users from sharing information necessary to perform their jobs.

3. Firewall and router administration requires careful and correct implementation of rules (system specific policy).

C. Port Protection Devices

A port protection device (PPD) is fitted to a communications port of a host computer and authorizes access to the port itself, prior to and independent of the computer's own access control functions. A PPD can be a separate device in the communications stream,² or it may be incorporated into a communications device (e.g., a modem). PPDs typically require a separate authenticator, such as a password, in order to access the communications port.

One of the most common PPDs is the *dial-back modem*. A typical dial-back modem sequence follows: a user calls the dial-back modem and enters a password. The modem hangs up on the user and performs a table lookup for the password provided. If the password is found, the modem places a return call to the user (at a previously specified number) to initiate the session. The return call itself also helps to protect against the use of lost or compromised accounts. This is, however, not always the case. Malicious hackers can use such advance functions as call forwarding to reroute calls.

Security Issues for Data Transfer

In addition to intruders possibly gaining access to internal systems, it is also possible to eavesdrop on an entire session. Eavesdropping is not technically difficult if there is physical access to cable or wire used for communication or logical access to switching equipment.

If a telecommuting employee will be transferring data for which someone would go to the trouble of eavesdropping to get, then encryption may be necessary. Another scenario when eavesdropping is more likely is if an employee is at a large conference or other location where an eavesdropper may set up equipment in hopes of hearing something useful. Some conferences offer equipment to attendees to use to check email, transfer files, etc. This is useful to attendees, since they do not need to provide laptops; however, this could be a target for electronic eavesdropping.

Software- or hardware-based encryption provides strong protection against electronic eavesdropping. However, it is more expensive (in initial and operating costs) than robust authentication. It is most useful if highly confidential³ data needs to be transmitted or if even moderately confidential data will be transmitted in a high threat area.

It is, however, unlikely that employees will always know when they are in a high threat area. It is incumbent on management to train employees.

² Typically PPDs are found only in serial communications streams.

³ Highly confidential implies that someone would actively pursue obtaining the data.

Security Issues for Telecommuting from Home

What this paper has discussed so far are issues related to protecting internal corporate systems and data in transit. Many employees telecommute from home, which raises an additional set of issues. Some of these concerns relate to whether employees are using their own computers or using computers supplied to them by the organization.

A. Home Data Storage Integrity and Confidentiality

Other members of the employee's household may wish to use the computer used for telecommuting. Children, spouses, or other household members may inadvertently corrupt files, introduce viruses, or snoop. Organizations can take several approaches:

1. Employee accountability. Some organizations may choose not to have specific rules forbidding household members from using PCs, but hold the employee responsible for the integrity and confidentiality of the data. Obviously, this is not a good choice if the data is highly confidential.
2. Removable hard drives. If corporate data is stored on a removable hard drive (or floppy), then the risk is greatly reduced.
3. Data encryption. Corporate data can be kept encrypted on the hard disk. This will protect its confidentiality and will detect changes to files.
4. Dedicated use. If an organization requires this, it should recognize that it is difficult to enforce.

B. Home System Availability

In addition to the possibility of a home computer breaking or being stolen, it may not be compatible with office configurations. For example, the home computer may use a different operating system. This may complicate set up, software support, troubleshooting, or repair. It is in the best interest of the organization to ensure that policy covers all these situations.

Security Issues for Telecommuting Centers

Telecommuting centers, normally located in outlying suburbs, are another choice for organizations. From a security perspective they may offer hardware for encryption, removable hard drives, and increased availability. However, by concentrating telecommuters, they may make themselves a more attractive target for eavesdropping. At a minimum, organizations should require robust authentication from telecommuting centers.

If communications encryption is supported by the center, organization should be aware that data may not be encrypted while it is inside the center. The encryption may occur at a modem pool.

Conclusion

In conclusion, telecommuting offers many benefits. With adequate attention to security, it is possible to create "an office away from the office."

References

Ascend Communications, **Telecommuting Network Planning Guide: A Resource Guide for Planners, Executives and Information Managers**, Alameda, CA.

Bill Boyle, *Cable & Wireless Staff are to Work from Home*, Computer Weekly, April 27, 1995, p6(1).

IDC Government, **Telecommuting: New Challenges in Information Security**, IDC G Pub. No.: W1831, March 1995.

NIST's Information Infrastructure Task Force Committee on Applications and Technology, **The Information Infrastructure: Reaching Society's Goals**, NIST Special Publication 868.

John Pescatore, *Telecommuting and Security Aspects*, Research Activity #9008, IDC Government, February 9, 1996.

Johna Till Johnson and K. Tolly, *The Safety Catch*, Data Communications Magazine, May 1995.

John P. Wack, and L. Carnahan, **Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls**, NIST Special Publication 800-10, December 1994.

<http://www.telecommute.org/links.html#tc> - includes resource links to new stories, organizations, teleworking studies, and telecommuting centers.

<http://www.pacbell.com/Lib/TCGuide/tc-12.html> - contains Pacific Bell's 4 page Telecommuting and Resource Access Security Checklist of questions to consider when creating a telecommuting security policy.

An Isolated Network for Research

Matt Bishop, L. Todd Heberlein
Department of Computer Science
University of California at Davis
Davis, CA 95616-8562

Abstract. An isolated network is critical to the successful analysis of vulnerabilities and attack tools. Maintaining such a network introduces issues of policy and implementation which conflict with the need to transport data from the Internet to the network. This paper describes the goals of one isolated network, the policy and implementation that satisfies those goals, and other considerations to protect the confidentiality of data and programs on the isolated network.

Keywords. Isolated network, vulnerability, attack tools, design, implementation

1. Introduction

Vulnerabilities research requires the analysis of known security problems, and the development of techniques and technology to find previously unknown security problems. Exercising vulnerabilities helps researchers understand how and why those vulnerabilities occur, what interrelationships with system components exist, and the effect of proposed patches. Ideally, one could perform this analysis from a description of the vulnerability; in practice, the complexity of modern systems makes such analysis merely a starting point.

Amplifying this is the observation that many vulnerabilities come to the attention of the research community when attackers exploit those holes. This phenomenon may have many explanations. That attackers exploit vulnerabilities provides a fruitful source of information about vulnerabilities, for attackers often leave behind attack tools (binaries or scripts) automating these exploitations. Executing these scripts, and analyzing their behavior, often makes determining which vulnerabilities the scripts exploit much easier.

Research in the area of computer and network vulnerabilities entails handling information which provides detail about specific methods to compromise the security of computer systems of a specific type. Worse, some data specifies sites, hosts, and even IP addresses on which the compromise occurred. This data is therefore considered sensitive, and must not be divulged to unauthorized users or made accessible to the Internet.

This paper discusses the model used to protect the information on an isolated network, how the model and network are implemented, and plans for extending the model. Section 2 presents the model and the reason for its selection. Section 3

discusses the implementation of the model in the setup and maintenance procedures for the isolated network, and other principles used to assure a reasonable level of security. Section 4 explains how data is moved to and from the isolated network, as such motion contradicts the notion of “isolation” and is a potential point of vulnerability. Section 5 explains the larger goals of the isolated network, and how its configuration serves to advance those goals.

2. Model

The isolated network (called “isonet”) currently consists of several hosts running different versions of the UNIX operating system (SunOS, Solaris, IRIX, and HP/UX). This constrained our choice of model, because the model had to be simple enough to be implemented using native UNIX protection mechanisms and yet powerful enough to provide adequate security.

“Adequate security” is a function of the isonet’s requirements, which are driven by the four types of data and programs that the isonet stores. The vulnerabilities database contains descriptions of the vulnerabilities, including system types on which they were found, environmental conditions needed to exploit them, and at least one attack script demonstrating how to exploit the problem. Attack tools recovered from many sources often show previously unknown vulnerabilities, and provide a basis for studying techniques for attack script analysis. Other researchers use the isonet as a testbed for security-related tools and protocols, such as comparing the effectiveness of different intrusion detection systems. Finally, the testbed provides a development environment for tools deemed too sensitive to be placed on the Internet, such as network connection altering tools.

The isonet must protect the confidentiality of data and programs stored on the isonet (called “isoinfo”) by providing the following:

1. a mechanism to keep the isoinfo inaccessible to users on the Internet; and
2. a mechanism to ensure authorized users can access only the type of information they are authorized to access.

These requirements immediately suggest a multi-level security model. That the systems are UNIX-based suggests one security level for the isonet. The hosts on the isonet all run at the High level and the Internet is considered Low; this division is enforced by physical means. The four categories are proprietary programs (Prop), development tools (Dev), vulnerabilities data, and attack tools; but as attack tools are part of the vulnerabilities studies, those two categories are merged (Vuln).

The Bell-LaPadula model [1] allows subjects in compartments to write to objects in the same compartment. This presents a problem: when attack tools are executed and vulnerability exploits are recreated, the exploitation could alter information under study or programs under development. So the model must be modified

subjects	objects			
	Low	(High, Vuln)	(High, Prop)	(High, Dev)
Low	rw			
(High, Vuln)		r		
(High, Prop)			r	
(High, Dev)				r
(High, AVuln)		rw		
(High, AProp)			rw	
(High, ADev)				rw

Figure 1. Access Matrix for the Isonet Model.

to provide:

3. a mechanism to ensure that the execution of a process does not affect isoinfo unless it is authorized to do so.

Lipner [2] showed how to extend the traditional MLS model to provide such a mechanism. Three new categories, AVuln, AProp, and ADev, allow member subjects to alter data or programs in the categories Vuln, Prop, and Dev, respectively. No objects reside in the new categories. The resulting access matrix is shown in Figure 1.

We show this configuration meets the above requirements.

Consider requirement 1. If the isonet is at level *High*, and the Internet at level *Low*, this bars writing between the isonet and the Internet as no subject at level *High* can write to an object at level *Low*.

Now consider requirement 2. Isonet users (subjects) are assigned to a set of categories corresponding to their needs. Subjects with a particular set of categories can only read objects within those categories to which they are assigned. Further, they can only alter information if they are in the appropriate "A" category; this restricts the ability of a user to damage data or programs when experimenting.

Clearly the security levels form a linear hierarchy. A subject in category ADev can read or write an object in Dev, but a subject in Dev can only read an object in Dev. This induces a relationship based on the number of rights a subject has over an object; defining the relationship in the obvious way, Dev < ADev, Prop < AProp, and Vuln < AVuln. This gives a lattice model of security, and meets requirement 3.

3. Implementation

Our implementation combines both procedural and technical mechanisms to achieve a level of security that prevents the accidental release of, and damage to,

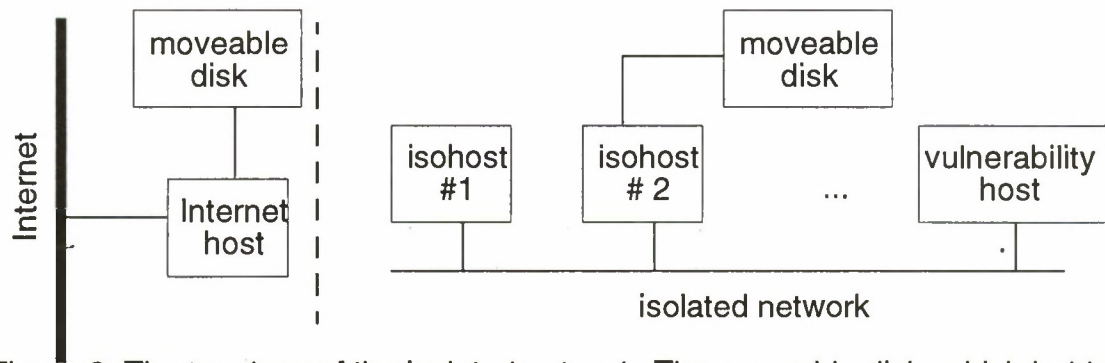


Figure 2. The topology of the isolated network. The moveable disk, which holds only data, is physically moved from the Internet host to an isohost to upload files, and from the isohost to the Internet host to download files.

the isoinfo, as well as to prevent external attacks. As is any site, the isonet is vulnerable to insider attacks; however, we have taken some steps to limit their damage and ensure they are quickly detected. This results from the systems not implementing multi-level security, and hence the isonet can only emulate the model, not implement it fully.

3.1 Configuration

Figure 2 shows the isonet topology. As the isonet grows, and the topology becomes more complex, routers, bridges, and other network infrastructure components will be added. However, the part to the left of the dashed line will remain unchanged. The configuration to the left is the interface to the Internet; the part to the right is the fully isolated net. The moveable disk is used to move data from the Internet to the isolated network, and vice versa, and will be discussed later.

The hosts labelled *isohost* are the hosts on the isonet. The isohosts contain the user home directories and non-sensitive programs (development tools that could be made available to the Internet, such as compilers, mailers, word processors, and so forth). The isohosts trust one another, and may use DNS and NIS or static host and user information tables. If the former, the isohosts are configured so that only the isonet DNS and NIS servers will be queried; if they fail, static information is used. All isohosts run remote login and file transfer servers; they may run other servers as well. Of course, these servers are inaccessible to the Internet.

One of the isohosts, the *vulnerability host*, stores vulnerability and attack information, as well as tools and programs deemed sensitive (such as proprietary programs or vulnerability detection or exploitation tools). If a user wishes to add a vulnerability or alter a program, that user must do so from this host. The vulnerability host is *never* used for experiments, and may be taken off the isolated network during experiments (should the experimenter deem it necessary).

3.2 Implementation of the Model

The separation between security levels is enforced physically; the isonet is disconnected from the Internet. This disallows High subjects from accessing Low objects, and vice versa, as the model requires.

Implementing categories is a bifurcated procedure. Users in categories Vuln or AVuln are in the group *vuln*; categories Dev and ADev correspond to *dev*, and Prop and AProp to *prop*. For example, user *bishop* is a member of the groups *vuln* and *dev*, and so can read source code for tools under development, and data in the vulnerabilities database; but this user cannot access proprietary programs.

The programs and data which this scheme protects reside on a disk exported from the vulnerability host. Each file on this disk is in one of those three classes, and all are group readable and writable; however, the disk is exported as read-only. So remote clients cannot alter the files, despite those files being group writable. The programs and data are kept unreadable and unwriteable by all other users (except the owner) to prevent any unauthorized access.

Subject membership in the categories AVuln, ADev, and AProp require that the user have an account on the vulnerability host. Then the users can alter the files, as the disk is local to the vulnerability host and is readable and writable there.

One advantage of these procedures is that no locally-developed software is required. Exporting is controlled using vendor-implemented protocols, currently the Network File System [4], and each vendor supplies its own account management facility.

While procedures clearly dictate that the isonet is never to be connected to the Internet, not all mistakes can be prevented. Hence the vulnerability host does not run any network servers other than the file exporting servers, and these are configured to export only to other isohosts. For example, no remote logins or file transfers are allowed. Second, the IP address of the vulnerability host is the same as the IP address of one of the Department's busiest servers. The vulnerability host is also a much faster system. Thus, should the isonet ever be connected to the Internet, it will receive messages intended for the Department server, and refuse them as it runs none of the daemons that the Department server makes available. Experience shows that users will quickly report these problems to the system staff, who can take immediate corrective action.

3.3 Other Aspects

Because systems are often altered for testing, the isonet hosts are maintained as close to the vendor-distributed configuration as possible. This makes reinitializing hosts very simple, and eliminates the problem of porting locally developed configuration management code to new types of systems as they are hooked up to the network. Two aspects of this are worth some elaboration.

Identification and authentication arise twice: first, in the issuing of new accounts, and second, in the access procedures. Accounts are under the control of the Computer Security Laboratory, so only those who have a legitimate need to access the isonet will receive accounts on it. To log in, a user must first enter a locked room (authorized users and graduate students have keys), and then go through the system authentication mechanism. Additional mechanisms have not proven necessary.

While auditing would allow tracking of uses of the isonet hosts, it would also provide no barrier to a determined insider. Further, providing a robust audit mechanism in this environment would require extensive system modifications. Given these, relying on the standard system audit mechanisms seemed appropriate.

3.4 Analysis

Saltzer and Schroeder's principles of secure design [3] are a useful metric against which to evaluate this design. The principles they enunciate are:

1. Users (processes) should have the minimum set of access rights necessary (least privilege).
Procedurally, users are assigned to the least upper bound of the compartments they must access to complete their tasks. The security mechanisms ensure that access to isoinfo is limited to those authorized by the class membership.
2. Access requires explicit permission (fail-safe defaults).
By default, users are members of the generic *other* group. Membership in a group must be given explicitly.
3. Design of the security mechanisms should be simple and small enough to be verified (economy of mechanism).
The model presented above cannot be simplified any further; the analysis in the previous section shows the model meets the stated requirements. The analysis in this section demonstrates that the model is implemented correctly.
4. Every access should be checked for authorization (complete mediation).
The UNIX operating system does not fully enforce this principle, checking authorization only when files are opened. But the above implementation meets this criterion to the extent possible.
5. Security should not depend on the secrecy of the design (open design).
The model and its implementation are available to all users.
6. Access to objects should depend on more than one condition being satisfied (separation of mechanism).
Access to the isonet itself requires physical access to a graduate student laboratory that is kept locked. Access to the data or to programs requires both an account and group membership; if the access enables the user to alter either data or programs, an account on one specific host is also required. Thus, several conditions must be met before access to objects is allowed.

7. Mechanisms shared by multiple users should be minimized (least common mechanism).

This principle cannot be enforced adequately because the isonet provides common hardware and software. In particular, if the owner of data in a particular compartment desires to make that data available to all others, a simple change of protection mode accomplishes this. Having trusted users own as many files ameliorates this considerably, but the threat is not eliminated.

8. Mechanisms must be easy to use (psychological acceptability).

The implementation mechanisms are standard UNIX security and system administration mechanisms and so are familiar to all our users. They require no extra software or hardware and thus are likely to be applied correctly and entail little to no extra burden on the users.

Thus, both the model and the implementation of the model meets the principles of secure design as much as possible in the UNIX environment

Because data from the Internet (and other sources) resides on the isonet, and data and programs are added as they become available, the above model must be modified to include the motion of data to and from the isolated network. The next section describes the modifications to the model and the implementation of a solution to this problem.

4. Uploading and Downloading

As research into vulnerabilities began, many helpful users and system administrators offered copies of various attack tools found at their sites. They enciphered these scripts using PGP [5] and mailed them to the first author, who moved the letters to a Macintosh, deciphered them, and put the cleartext onto floppies. The cleartext attack Tools could then be placed on the isonet.

Augmenting the model to handle the reclassification of data from Low to High would imply that users in the compartments would be able to read and write to the Internet. This is undesirable for several reasons. First, the users may download sensitive information without meaning to, for example by mistyping a file name. Second, users may upload programs with Trojan horses or other malicious logic. Third, if a user of the isonet wishes to pass information to someone on the Internet, or to use an attack script against an Internet host, the user may download the information or the attack script. Thus, access to the Internet is an exception to the rules of the model. This emphasizes the trust in those granted such access.

The moveable disk is a disk that can be connected to either a system on the Internet or a system on the isonet, but not both simultaneously. Because this is the *only* hardware that can be on both the Internet and the isonet, its management is central to the security of the vulnerability and attack data. Its requirements are:

1. The structure of the isonet must not be visible from the Internet. This allows the isonet to be reconfigured, and network infrastructure added, without affecting any other hosts or databases.
2. The moveable disk cannot contain any programs other than those being transferred. In particular, no system or user binaries may reside on it.
3. The moveable disk must not be a networked disk. Uploading data to, or downloading data from, the Internet host requires the user to be physically at the Internet host, to which the moveable disk hardware is attached.
4. The moveable disk must hold no sensitive data (except, possibly, for the data being uploaded or downloaded). This includes cryptographic keys.

The moveable disk requires special-purpose hardware (basically, a mounting bay) and so can only be used on systems with that hardware installed. This allows tight control over which hosts can be accessed, but requires a two-step process to upload data (the data must be placed on the moveable disk attached to the Internet host, and then that disk physically moved to an isohost and the data transferred).

Figure 2 shows the relationship of the moveable disk to the isonet and the Internet. The management procedures and hardware set-up implement the above requirements directly. Further, only those users trusted to upload or download data or programs have accounts on the Internet host; in other words, accounts on this host are completely independent of the accounts on the other isonet hosts. If a user without an account on the Internet host wishes to move data between the isonet and the Internet, a trusted user must perform the transfer; as trusted users will question the need for such a transfer, this performs the function of a trusted certification that the data may indeed be transferred without endangering confidentiality.

Remote users and system administrators who wish to contribute data, programs, or attack tools are rarely willing to send the information over the Internet in the clear. To allow the information to be sent in encrypted form, a PGP key pair is associated with the isonet. The secret key resides on the vulnerability host. The public key is available on a number of public servers, and may be distributed freely. Contributors of information may use this key to encipher the data before they send it; as the secret key resides on the isonet, the recipients can only decipher the contribution on that network.

5. Conclusion

The isolated network has been in use for two years, and the current model evolved from our experiences and the needs of our laboratory and contributors. It appears to work quite well, as we have not yet had a leak of information from the isonet. This also speaks of the character of the users of the isonet, and emphasizes the need for non-technical controls. Undoubtedly, as the uses and needs of

the project change, the model will evolve; in particular, network infrastructure will be added to enable us to test network-based vulnerabilities.

The isonet is a component of the Information Warfare Forensic Center. The IWFC's mission is to study the nature and types of vulnerabilities in complex systems including operating systems, network applications, and the network infrastructure (such as DNS, routers and their protocols). Among its goals are an understanding of why vulnerabilities occur, how to prevent and detect them, how to detect exploitation of vulnerabilities, and how to classify vulnerabilities. The development of vulnerability models is central to meeting these goals. Another primary objective is to develop forensic tools and methodologies to detect, analyze, and counter attacks. These tools will provide the foundation with which we can observe and analyze vulnerability exploitation and their effects in progress. The isolated network provides the foundation for experiments in support of these goals.

Acknowledgements: This work was supported under a contractual arrangement with the United States Air Force. Our thanks to our sponsors, especially Kevin Ziese and Scott Waddell, and to those who have helped build, and rebuild the isolated network, especially David O'Brien, Michael Dilger, and Scot Templeton.

6. References

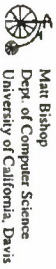
- [1] D. Bell and L. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model," Technical Report M74-244, MITRE, Bedford, MA (Oct. 1974).
- [2] S. Lipner, "Non-Discretionary Controls for Commercial Applications," *Proceedings of the 1982 Symposium on Security and Privacy* pp. 2-10 (Apr. 1982).
- [3] J. Saltzer and M. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE* **63**(9) pp. 1278-1308 (Sep. 1975)
- [4] Sun Microsystems, Inc., "NFS: Network File System Protocol," RFC 1094 (March 1989).
- [5] P. Zimmerman, *PGP User's Guide* (Sep. 1992).

An Isolated Network for Research

Matt Bishop, L. Todd Heberlein
Department of Computer Science
University of California at Davis
Davis, CA 95616-8562

electronic mail: bishop@cs.ucdavis.edu
phone: (916) 752-8060

acknowledgment: this work funded by a contract to the University of California,
Davis from the United States Air Force



Slide # 1

Outline

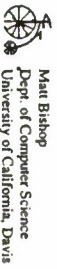
- Goal
- Policy Issues and Model
- Implementation Issues
 - configuration
 - enforcement of separation
 - other aspects of the model
 - analysis of effectiveness
- Uploading and Downloading



Slide # 2

Goals

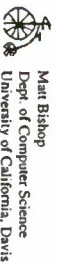
- To provide an environment for:
- studying vulnerabilities
 - testing fixes to vulnerabilities
 - analyzing attack tools



Slide # 3

Policy Issues

- Keep data on the isonet inaccessible except to authorized users
 - Some data/tools come from public sources; others don't
- Limit access of authorized users to the data they are authorized to use
 - Separate tool testers from vulnerability data
- Execution of a process should not affect isonet unless it is authorized to do so
 - don't let attack tools mess up databases



Slide # 4

Model

Use multilevel security model

- Low
 - Internet
- High
 - Vuln, Prop, Dev: categories for read access
 - AVuln, AProp, ADev: categories for alter (read/write) access

Isonet users assigned to appropriate compartment(s)



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 5

Implementation Issues

- Procedural
 - Keep isonet physically separated from Internet
 - Keep vulnerabilities repository, attack scripts on vulnerability host
 - Vulnerability host has IP address that will cause problems
 - Limit logins to those with reasons for access
 - Isohost configuration information not given to Internet hosts
 - Identification and authorization controlled by isonet folks, not department



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 6

Implementation Issues (con't)

- Technical
 - Use NFS restrictions to control read, write access to files
 - Regular (non A-) compartments on any isohost
 - All sensitive information kept on vulnerabilities host
 - All isohosts can mount vulnerabilities host file systems, but read-only
 - Files owned by groups *dev*, *prop*, *vuln* and readable/writable only by them
 - To write (i.e., enter A- compartments), need login on vulnerabilities host
 - Use standard (vendor-supplied) mechanisms throughout



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 7

Analysis

- Implements Saltzer's and Schroeder's Principles of Secure Design as much as feasible in the given environment
- Vulnerable to trusted insider attack
 - But what isn't?

1



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 8

Internet Interaction

- Need to upload information from Internet
 - People send attack tools, vulnerability information, etc.
 - May need to download analyses (enciphered, of course)
- Technique
 - First considered bridge host that could be on one or the other network
 - Simpler: moveable disk



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 9

Moveable Disk Requirements

- Contains no information about structure of isonet
 - Strictly for ferrying data, etc. between isonet, Internet
 - *Never* used for anything else, *especially* sensitive data
- No programs unless they are being transferred
 - Disk is untrusted
 - Disk is cleaned after each transport
- Not exported on network
 - To read from or write to it, you need to be on the Internet (isonet) host to which it is physically connected (think separation of privilege)



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 10

Transferring Data Over the Internet

- Why?
 - Contributions from others
 - Sending out analyses
- Use PGP (or some other encryption mechanism)
 - Public key is publicly available
 - Private key kept on vulnerabilities host



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 11

Information Warfare Forensics Center

To study the nature and types of vulnerabilities in complex systems such as operating systems, network applications, network infrastructure (DNS, routers, etc.)

- why do vulnerabilities occur?
- how can they be prevented and/or detected?
- how can their exploitation be detected?
- how can they be classified to show their interrelationships?

Other goals:

- develop forensic tools, methodologies to detect, analyze, and counter attacks



Matt Bishop
Dept. of Computer Science
University of California, Davis

Slide # 12

GrIDS—A GRAPH BASED INTRUSION DETECTION SYSTEM FOR LARGE NETWORKS*

S. Staniford-Chen, S. Cheung, R. Crawford, M. Dilger,
J. Frank, J. Hoagland, K. Levitt, C. Wee, R. Yip, D. Zerkle

*Department of Computer Science,
University of California, Davis,
Davis, CA 95616*

email: <lastname>@cs.ucdavis.edu

Abstract

There is widespread concern that large-scale malicious attacks on computer networks could cause serious disruption to network services. We present the design of GrIDS (Graph-Based Intrusion Detection System). GrIDS collects data about activity on computers and network traffic between them. It aggregates this information into activity graphs which reveal the causal structure of network activity. This allows large-scale automated or co-ordinated attacks to be detected in near real-time. In addition, GrIDS allows network administrators to state policies specifying which users may use particular services of individual hosts or groups of hosts. By analyzing the characteristics of the activity graphs, GrIDS detects and reports violations of the stated policy. GrIDS uses a hierarchical reduction scheme for the graph construction, which allows it to scale to large networks. An early prototype of GrIDS has successfully detected a worm attack.

Keywords: Intrusion detection, networks, information warfare, computer security, graphs.

1 Introduction

The Internet is increasingly important as the vehicle for global electronic commerce. Many organizations also use Internet TCP/IP protocols to build

intra-networks (intranets) to share and disseminate internal information. A large scale attack on these networks can cripple important world-wide Internet operations. The Internet Worm of 1988 caused the Internet to be unavailable for about five days [1]. Seven years later, there is no system to detect or analyze such a problem on an Internet-wide scale. The development of a secure infrastructure to defend the Internet and other networks is a major challenge.

In this paper, we present the design of the *Graph-based Intrusion Detection System* (GrIDS). GrIDS' design goal is to analyze network activity on TCP/IP networks with up to several thousand hosts. Its primary function is to detect and analyze large-scale attacks, although it also has the capability of detecting intrusions on individual hosts. GrIDS aggregates network activity of interest into *activity graphs*, which are evaluated and possibly reported to a system security officer (SSO). The hierarchical architecture of GrIDS allows it to scale to large networks.

GrIDS is being designed and built by the authors using formal consensus decision-making and a well-documented software process. We have completed the GrIDS design and have almost finished building a prototype.

This paper is organized as follows. Section 1.1 briefly describes related work on intrusion detection systems and motivates the need for GrIDS. Section 1.2 discusses classes of attacks that we expect to detect. In Section 2.1, the simple GrIDS detection algorithm is described, followed by a more detailed

*The work reported here is supported by DARPA under contract DOD/DABT 63-93-C-0045.

discussion in Section 2.3. Section 2.4 has a treatment of the hierarchical approach to scalability and Section 2.5 discusses how the hierarchy is managed. Section 2.6 outlines the policy language. Section 2.7 covers some limitations of GrIDS. Finally, Section 3 presents conclusions and discusses future work.

1.1 Previous Work

The field of intrusion detection began with a report by Anderson [2] followed by Denning's well-known paper that became the foundation for IDES [3]. A recent review of the field is available [4] that gives more detail than we can provide here.

Early systems were designed to detect attacks upon a single host (e.g., IDES (later NIDES) [5, 6] and MIDAS [7]). Although they could collect reports on a single local area network (LAN), these systems did not aggregate information on a wider scale.

Later systems considered the role of networks. The Network Security Monitor (now called Network Intrusion Detector or NID) looked for evidence of intrusions by passively monitoring a single LAN [8]. NADIR [9] and DIDS [10] collect and aggregate audit data from a number of hosts to detect co-ordinated attacks against a set of hosts. However, in all cases, there is no real analysis of patterns of network activity; aggregation is used only to track users that employ several account names as they move around in the network.

NADIR and DIDS use distributed audit trail collection and centralized analysis. Centralized analysis severely limits the scalability of the detection algorithms. In internetworks of multiple administrative domains, different domains may be unwilling to share all activity information with others. Also, sufficient processing and communications resources to analyze activity in very large internetworks is unlikely to be available.

GrIDS moves beyond these limitations by using a hierarchical aggregation scheme in order to scale to larger networks.

1.2 Network Attacks

This section briefly discusses some large-scale attacks that GrIDS aims to detect; it indicates how GrIDS distinguishes malicious activities from normal behavior.

A *sweep* occurs when a single host systematically contacts many others in succession. *Doorknob rattling* is a sweep that checks for vulnerable hosts, (e.g. hosts that employ weak or default passwords on user accounts). There are legitimate reasons for sweep activity (e.g. polling of network resources such as SNMP, centralized backups, audit sweeps by security administrators). However, legitimate sweeps tend to be highly circumscribed and regular—the source host, services used, hosts contacted, and time of day are known. Thus, they can be differentiated from malicious sweeps.

Coordinated attacks are multi-step exploitations using parallel sessions where the distribution of steps between sessions is designed to obscure the unified nature of the attack or to allow the attack to proceed more quickly (e.g. several simultaneous sweep attacks from multiple sources). The combined nature of the distributed attack is only apparent if the attack is traced back to the same source, or if features of the attacks are similar. To detect such coordinated activity, an IDS must correlate sessions across several hosts and possibly across several distributed detectors.

Seely [11] defines a *worm* as “a program that propagates itself across a network using resources on one machine to attack other machines.” The best known worm attack is the Internet worm of 1988 which infected thousands of hosts throughout the Internet, rendering many of them unusable. Worms are evidenced by a large amount of traffic forming a tree-like pattern and by similar activity occurring amongst affected hosts. Intrusion detection systems may detect a worm by analyzing the pattern of spread.

2 GrIDS—Graph-Based Intrusion Detection System

We now explain the nature and operation of the GrIDS system. Firstly, we present a simple example to illustrate the main concept. Next, we discuss the architecture and components that make up the distributed system. Then we give a more detailed description of how these components operate to detect intrusions. For a complete account, refer to [12].

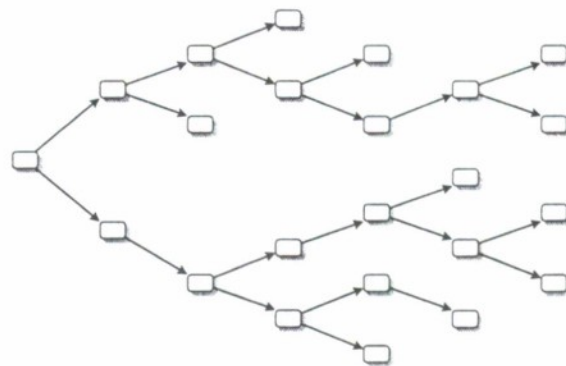
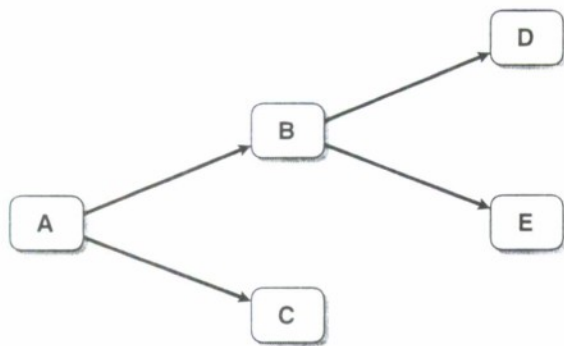


Figure 1: The beginning of a worm graph, and the graph after the worm has spread.

2.1 Detecting a Worm

GrIDS constructs *activity graphs* which represent hosts and activity in a network. Let us take the tracking of a worm as a simple example of building such an activity graph. In Figure 1, the worm begins on host *A*, then initiates connections to hosts *B* and *C* which causes them to be infected. The two connections are reported to GrIDS, which creates a new graph representing this activity and records when it occurred. The two connections are placed in the same graph because they are assumed to be related. In this case, this is because they overlap in the network topology and occur closely together in time.

If enough time passes without further activity from hosts *A*, *B*, or *C*, then the graph will be discarded. However, if the worm spreads quickly to hosts *D* and *E*, as in the figure, then this new activity is added to the graph and the graph's time stamp is updated. Eventually, the worm's spread is represented as a larger graph, as shown on the right of Figure 1.

Thus when a worm infects a network protected by GrIDS, the network activity associated with its propagation causes GrIDS to build a tree-like graph. A detection heuristic can recognize this tree-like graph as a potential worm. This evaluation might count the number of nodes and branches in the graph. Recognition (detection) occurs when the counts exceed a user-specified threshold, thus causing GrIDS to report a worm.

In the previous example, all connections were incorporated into the graph regardless of connection type. GrIDS can use other information to relate network activities, such as destination port numbers, or the type of operating systems. In fact, ar-

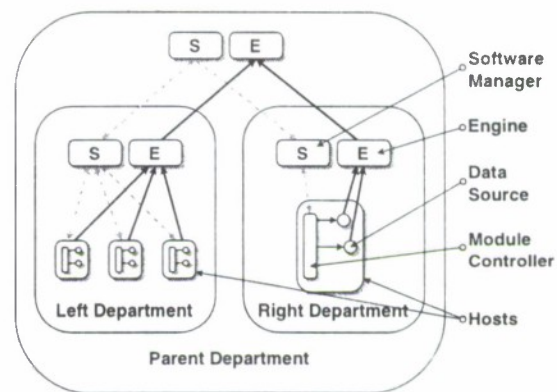


Figure 2: Overall architecture of the system.

bitrary information can be utilized since GrIDS can import user-supplied correlation functions into its graph building algorithm.

Similarly, sweeps and other patterns of misuse produce graphs of a characteristic shape, and GrIDS may be programmed to detect and report them.

To verify our design concept, a basic implementation of this algorithm (which we christened *Early Bird*) was built. While it would be premature to quantitatively evaluate this version, the code was tested for several weeks on our LAN with tcp-wrapper [13] data as input. It was not difficult to tune the software to detect a worm or sweep attack within seconds but produce only one or two false alarms per day from normal user traffic.

2.2 Architecture

Figure 2 depicts a simple hierarchy with three departments: *Left* has three hosts, *Right* has one host, and *Parent* contains *Left* and *Right*.

All GrIDS software is in the form of modules with a standardized interface. The modules are started, stopped, and controlled by a module controller process located on each host.

Each department has two special modules: the software manager and the graph engine. The software manager is responsible for managing the state of the hierarchy and the distributed modules. The hierarchy is re-arranged dynamically by drag-and-drop in a user interface, and starting and stopping particular modules is similarly automated.

GrIDS data sources are modules that monitor activity on hosts and networks and send reports of detected activity to the engine. The activity is reported in the form of a node or an edge for possible inclusion in an activity graph.

Data sources that are part of GrIDS include network sniffers and point IDSs (intrusion detection systems that work on a single host or LAN). However, GrIDS provides an extensible mechanism such that other security tools can be incorporated as data sources without significant change to the tool or GrIDS.

The graph engine takes input from data source modules. The engine builds graphs, and then passes summaries of those graphs up to the engine for its parent department. The parent engine, in turn, builds graphs which have a coarser resolution.

In addition to the components shown, there are user interface modules for allowing human interaction with the system, management functions, and display of alerts. There is also a central organizational hierarchy server which has a global view of the topology of the hierarchy, and is responsible for ensuring that changes to the hierarchy happen in a consistent manner.

2.3 Graph Building

This section discusses the GrIDS *engine*, which collects reports from the data sources and builds them into graphs.

Graphs consist of nodes and directed edges. A single graph represents a causally connected set of events on the network. Nodes represent hosts or departments, and edges represent network traffic between them. Nodes and edges are annotated with attributes that hold supplementary information. In addition, a graph has *global attributes* which maintain state information about the graph as a whole.

Because GrIDS searches for numerous types of

network abuse, different kinds of graph are needed. Graphs are constructed in a flexible way; users write *rule sets* which specify how graphs are built from reports. A single graph containing all network activity is too awkward to analyze effectively, so GrIDS allows multiple rule sets. For each rule set it maintains a *graph space* which contains a number of connected graphs. A rule set is an executable specification of one kind of graph; it determines whether an incoming report will be incorporated into existing graphs, and what the results will be. It also specifies when the engine will consider a graph as suspicious and what actions to take if it is. Rule sets operate independently from one another.

Each new report is presented to each rule set in the form of a partial graph. If the report satisfies the rule set's *preconditions*, the engine considers adding the report to the graphs in that rule set's graph space.

A rule set specifies *combining rules* (for nodes and for edges), to determine if an incoming graph should be combined with an existing overlapping graph, and how that should occur. Disjoint graphs cannot be combined. If a combining condition is satisfied on at least one node or edge, then the incoming graph is combined with that existing graph, and the graph's attributes are recomputed. Finally, if no graph combining occurs, but the incoming report did pass the preconditions, then it forms a new graph in the graph space.

2.3.1 An Example Rule Set

Rule sets serve several purposes: to decide if two graphs should combine, to compute the attributes of the combined graph, and to decide what actions to take, if any. Computing the edges and nodes in the combined graph is a straightforward matter which the engine does automatically. However, since it does not know the semantics of user-defined attributes, the rule set must specify how to combine them

A rule set consists of several sections:

- A name
- Initializations
- Preconditions
- Graph combining rules
- Assessment and actions

The following example rule set detects worms by aggregating adjacent connections into the same graph if they occur close together in time. It also includes any node reports which have an *alert* attribute, if they fall in the appropriate time frame. Some portions of the rule set which give low level detail have been omitted for clarity.

Throughout the following rules, *new* refers to attributes appearing on the incoming report, *cur* refers to attributes appearing on an existing graph for this rule set, and *res* refers to attributes being computed for the resulting graph. The { ... } syntax denotes a set constructor.

```
ruleset worm_detector;
```

```
timeout 30;
```

```
report global rules {
  res.global.alerts = {};
  res.global.time = 0;
}
```

```
node precondition defined(new.node.time)
    && defined(new.node.alert);
edge precondition defined(new.edge.time);
```

The report global rules initialize the graph space.

Node and edge preconditions filter the reports that are not pertinent to the kind of abuse that this rule set is trying to detect. For each node and edge in the incoming graph, the appropriate kind of precondition is evaluated.

This node precondition requires an incoming node to have a *time* attribute and an *alert* attribute. Similarly, incoming edges (in reports) are accepted if they possess a *time* attribute.

Node rules may access both sets of global attributes and the attributes on the local node being considered. The sample adds any *alert* attributes on the current node to the global *alerts* attribute, initializes the local *alerts* attribute and the *time* attribute. Similarly, the edge rules combine alerts.

```
report node rules {
  res.global.alerts =
    {res.global.alerts, new.node.alert};
  res.node.alerts = {new.node.alert};
  res.global.time =
    max({res.global.time, new.node.time});
```

```
    res.node.time = new.node.time;
}

report edge rules {
  res.global.alerts =
    {res.global.alerts, new.edge.alert};
  res.edge.alerts =
    {res.edge.alerts, new.edge.alert};
  res.global.time =
    max({res.global.time, new.edge.time,
        new.source.time, new.dest.time});
  res.edge.time = max({new.edge.time,
        new.source.time, new.dest.time});
}
```

The next three sections of the rule set specify whether to coalesce two graphs, and compute attributes on the coalesced graph. (Disjoint sub-graph global attributes are re-computed on those nodes and edges within the intersection of two graphs.)

First we specify how the global attributes of two disjoint sub-graphs are combined by the engine. This initial combination can be modified by subsequent local rules. The *combine global* section updates the global *alerts* attribute for a graph to be the union of the existing *alerts* attributes of for the graphs under combination:

```
combine global rules {
  res.global.alerts =
    {new.global.alerts,
     cur.global.alerts};
}
```

The attribute *combine* determines whether the graphs should be combined. If the *combine* attribute evaluates to true on *any* overlapping node or edge in the sub-graphs, then the graphs are coalesced. In the example below, the sub-graphs are combined if at least one of the shared nodes has a non-empty *alerts* attributes, and if the nodes' time attributes are within thirty seconds.

If the sub-graphs are combined, the remaining node rules specify how attributes at nodes combine. In this case, the *alerts* attribute at a node in the final graph is the union of the *alerts* attributes for the constituent nodes, and the *time* attribute is the latest of the *time* attributes on the constituent nodes.

The edge rules are similar.

```
combine node rules {
  res.node.combine =
```

```

    !empty({new.node.alerts,
            cur.node.alerts})
    && abs(cur.node.time -
            new.node.time) < 30;
    res.node.alerts = {cur.node.alerts,
                      new.node.alerts};
    res.node.time =
        max({cur.node.time, new.node.time});
}

combine edge rules {
    res.edge.combine =
        abs (cur.edge.time - new.edge.time)
        < 30;
    res.edge.alerts =
        {cur.edge.alerts, new.edge.alerts};
    res.edge.time =
        max({cur.edge.time, new.edge.time});
}

```

Finally, the assessment rules evaluate the resulting graph and take appropriate actions. The actions on the right hand side are built-in functions, user defined functions, or updates to global attributes.

```

assessments {
    (!empty(res.global.alerts)) ||
    (res.global.nnodes >= 8) ||
    (res.global.nedges >= 13) ==>
        alert(), report-graph();
    (3 < res.global.nnodes < 8) ||
    (5 < res.global.nedges < 13) ==>
        report-graph();
}

```

Note that several attributes referred to above were neither declared nor computed by the earlier rules. These are *automatically computed* attributes; their values can be read by the rules, but not written:

- **global.ruleset** – the name of the rule set.
- **global.nnodes** – the number of nodes in a graph.
- **global.nedges** – the number of edges in a graph.
- **node.name** – the name of this particular node.
- **edge.source** – a list of the domains associated with the source of this edge that are within this engine's domain, starting with the domain for the source within this engine's domain and ending with the host.

- **edge.dest** – same as **source** except pertaining to the destination side of the edge.

2.4 Aggregation

GrIDS models an organization as a hierarchy of departments and hosts. Each department in the hierarchy has an engine of its own, which builds and evaluates graphs of activity within that department. However, activity which crosses departmental boundaries is passed up to higher levels in the hierarchy for further analysis.

As graphs propagate upward, entire departments may be represented as a single vertex, rather than a vertex per host, in a *reduced graph*. For example, the graph in Figure 3 represents an activity that involves hosts of three departments. Each department sees only the activity within its boundaries; these do not appear suspicious. The whole graph is not visible from any of the lower departments.

The higher level department does not have access to the full graph on the left either. At this level in the departmental hierarchy, the reduced graph (shown on the right) is seen. Because some information has been lost in the reducing of the subgraphs, this graph's topology is not suspicious either. However, attributes of the individual subgraphs are passed up forming attributes on the nodes in the aggregated graph. This allows the higher level module to draw stronger conclusions about the graph.

For example, each sub-department can pass up the size of the subgraph it sees, the branching factor of the graph, and the entrance and exit points of the graph into and out of this department. Thus, GrIDS can deduce that the total graph seen at the higher level has ten hosts in it. Similarly, an approximation of the branching factor and the depth of the graph can be computed.

Intractably large graphs never appear at any level. At lower levels, only sections of the graph are seen. At higher levels, only summary information about lower graphs is seen. Using this approach of aggregating graphs, GrIDS infers and reduces the data that must be analyzed at the higher levels of the hierarchy. It is this that makes GrIDS a scalable design.

The hierarchy which handles aggregation of graphs is also used to manage rule sets. A rule set is inherited by all the descendants of that node.

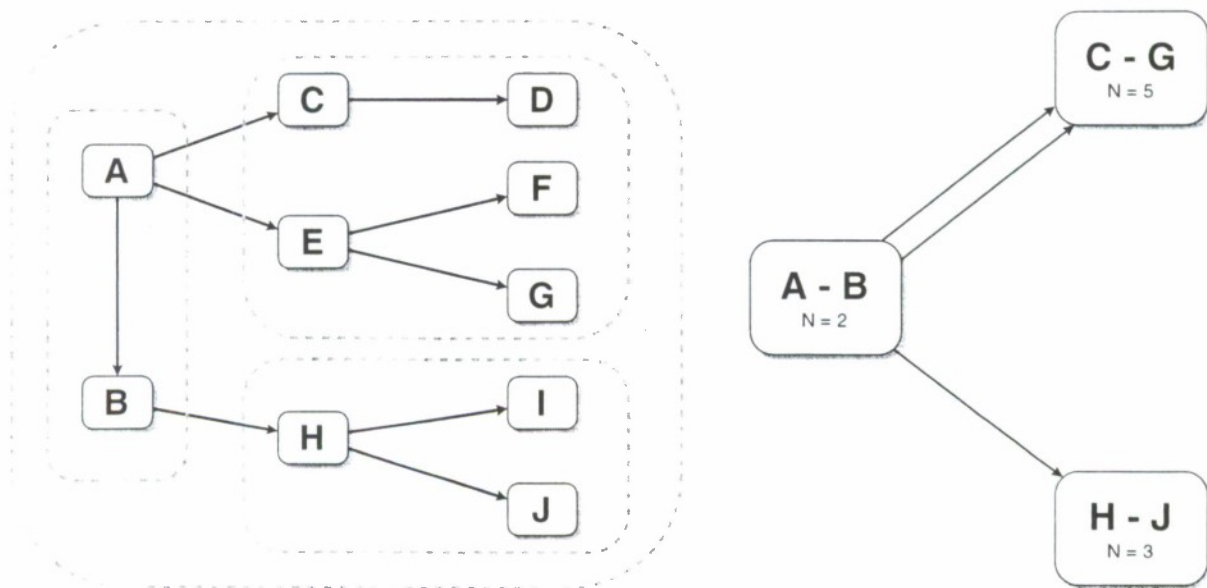


Figure 3: A graph amongst several departments (left), and the corresponding reduced graph. The dashed lines are departmental boundaries.

2.5 Managing the Hierarchy

Since organizations change, the hierarchical structure of departments and hosts must permit changes, but only by authorized users. This section describes how the hierarchy can be changed in a consistent manner.

An access control system controls who is able to view and manage the hierarchy. Each node (host or dept) in the hierarchy maintains an access control list (ACL) that specifies who may manage that node or any node in the subtree rooted there.

Users manage the hierarchy through *views* of subsets of the hierarchy which show the topology of the departments and hosts involved, and making *transactions* which change the hierarchy. Transactions include moving a department, adding a new host, changing the location of the graph engine for a department, etc. The challenge is to ensure that these transactions occur atomically and that the hierarchy is always left in a consistent state afterwards.

Several modules are involved in implementing the hierarchy. Each department has a software manager which is responsible for monitoring the hosts and modules in its department, tracking which hosts are currently functioning, and maintaining department-wide states such as the access control list. In addition, each host has a module controller responsible for the GrIDS software running on that particular

host. There are multiple user interfaces which have various views of parts of the hierarchy. All of these must be kept consistent.

Software managers and module controllers only know the *local* topology, i.e. their immediate parents and children.

A centralized *organizational hierarchy server* (OHS) maintains a complete global picture of the entire hierarchy. User interfaces maintain copies of as much of the hierarchy topology as their users presently wish (and are authorized) to manage.

Local knowledge simplifies efficient implementation of atomicity and consistency; locking, *etc.* can be centralized at the OHS. The use of a centralized system has some potential to limit scalability. Clearly, a single OHS will not work for the entire Internet. However, the OHS is only involved in *changes* to the topology of GrIDS, not in its routine operation. Hence, this limitation is not pressing.

We now outline how a transaction on the scenario depicted in Figure 4 would be carried out. Full details can be obtained from [12].

In the following, we use the notation S_C to refer to the software manager at C , M_C to refer to the module controller on the machine on which S_C is running, and similarly for the other departments. The organizational hierarchy server is O , and the interface is I .

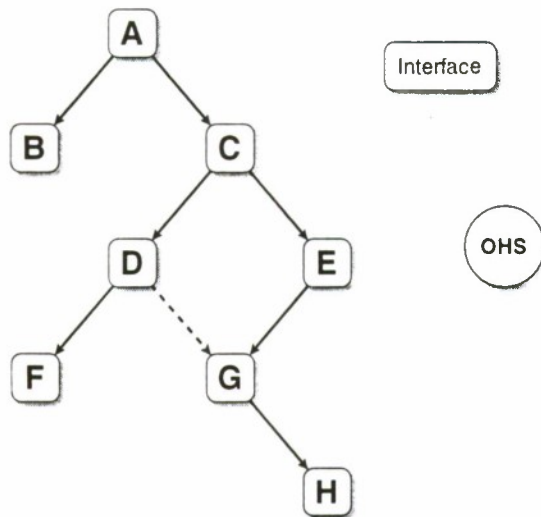


Figure 4: An example hierarchy. Department G is about to be moved from under department E to under department D. An interface and the organizational hierarchy server are also shown.

When a user starts up an instance of the user interface, she is prompted for a user identifier, a password, and a department in the hierarchy which she wishes to administer (in this case C').

I requests a copy of the hierarchy below C from O. O contacts S_C to verify that the user is authorized to access C. Then O replies to I with a copy of the hierarchy rooted at C'. O maintains a list of interfaces that have copies of the hierarchy. I displays the subtree on the user's screen. The copy is marked with a *version number* which is used in subsequent transactions to detect stale copies.

Suppose that, having inspected the hierarchy, the user decides to move department G (and by implication, its descendants) under D instead of E (illustrated by the dashed line in figure 4). The first step is to send a message to O. This message describes the planned action and supplies the hierarchy version number on which the planned action was based. O first determines if I's planned action is consistent with existing locks in the hierarchy and based on an up-to-date view. If so, it locks the appropriate part of the hierarchy and contacts S_D and S_E to verify that the user has the necessary permissions. If she does not, the lock is released. Assuming that the action appears feasible, O gives permission for I to go ahead.

Now I contacts S_E (the software manager for the

parent of department G) and informs it that G is to be moved. S_E sends messages to S_G warning of the impending change. Then S_E sends messages to M_G to alter the destination of S_G and A_G 's messages and the location of M_G 's parent. Since modules only have local knowledge, only S_G , A_G and M_G need to be updated. If these transactions have succeeded, S_E updates its own data structures and acknowledges completion to I.

Next I informs S_D of the move. S_D then informs S_G of the new information it needs to be a child of D (e.g., the access control list inherited from S_D). Upon completion, I reports back to O. O may then remove the locks on the hierarchy. Finally, O advises the interfaces that have invalid copies of the hierarchy. The OHS makes a best effort to inform the interfaces but does not block if interfaces are busy or no longer exist as this could prevent subsequent OHS transactions from proceeding. If any interface is not updated, the use of version numbers ensures that any transactions using stale hierarchy data are detected.

2.6 Policy

GrIDS includes a policy language to express acceptable and unacceptable behavior on the network. A network is a collection of users, hosts and departments. These entities communicate via pair-wise network connections which are labelled with the application protocol employed (e.g., TELNET, NFS, HTTP). Thus, a connection originates from a user, host or department and terminates in another user, host and/or department.

Policies are compiled into rule sets which build graphs and evaluate them for policy violations. This saves the user from having to write rule sets manually. In general, rule sets are more complicated to specify correctly than are policies. The present version of GrIDS only allows for policies stated with respect to a single graph edge (network connection).

The authorization model employed is similar to an access control model. The user specifies whether a connection is permitted or prohibited. Thus a rule regarding a certain type of connection consists of a tuple (*action*, *time*, *source*, *destination*, *protocol*, *stage*, *status*, ...) where *action* is allow or deny, *time* qualifies the rule with respect to a clock or time interval, *source*, and *destination* describe the connection endpoints and *protocol* describes the connection type. A connection progresses through several stages

(e.g. start, login, authentication, stop, etc.), and the *stage* and *status* attribute further characterizes the connection.

As an example, consider the policy

No student in the Computer Science Department is to read or write to the grade server hosted in Administration; faculty are permitted to submit grades and to read grades; teaching assistants are permitted to read grades; the department chair is permitted to change grades.

To check this policy with GrIDS, the policy compiler generates rule sets for three domains: Computer Science, Administration, and the department that constitutes the least upper bound of these two domains. The policy writer merely specifies the tuple that identifies which connections between these domains are allowed or disallowed.

Even though this policy mechanism is very simple, it allows considerably more flexibility than is possible with the main tool currently used for expressing network access policies: firewalls.

2.7 Limitations

GrIDS tackles some of the hard issues which need to be faced for an intrusion detection system to operate on a large network. A lot of our effort has gone into making the aggregation mechanism scalable, and allowing the system to be dynamically configurable so that it is still manageable when deployed on a large scale.

The current version of GrIDS is intended as a proof of concept for our approach to scalability and aggregation; as such, it has limitations. Before GrIDS can be considered for deployment in production environments, additional safeguards must be taken to ensure the integrity of communications between GrIDS modules, and to prevent an attacker from replacing parts of GrIDS with malicious software of her own. The prototype will not be resistant to denial of service attacks, disruptions of the network time protocol, or faults in the networks or computers on which it runs.

GrIDS is designed to detect large-scale attacks or violations of an explicit policy. A widespread attack that progresses slowly might not be diagnosed by our aggregation mechanism. However, suspicious activity associated with the attack could be detected

since point IDSs can be installed on GrIDS to detect intrusions that involve only one or a few sites.

3 Conclusions

We have presented the design of GrIDS. We have argued that GrIDS is helpful in detecting automated and spreading attacks on networks. GrIDS presents network activities to humans as highly comprehensible graphs. In addition, the GrIDS policy mechanisms allows organizations much greater control over the use of their networks than is possible, for example, with firewalls alone. GrIDS does this in a manner that is scalable and requires modest resources. GrIDS itself is manageable.

There is a great deal of further work to be done on GrIDS. The initial design is complete, and a prototype implementation is almost finished. We will proceed to evaluate the prototype and publish those results. Beyond that, robustness against random faults and attacks on GrIDS itself is the next priority. We also plan to further refine the policy language implemented by GrIDS.

Many important networks are vulnerable to widespread attack. We hope that GrIDS is a helpful step toward defending against such attacks.

Acknowledgements

We are grateful to DARPA for funding this research and to our technical monitor there, Teresa Lunt, for helpful discussion of this design.

References

- [1] M. Eichen and J. Rochis. With microscope and tweezers: An analysis of the Internet worm of November 1988. *IEEE Symposium on Research in Security and Privacy*, 1989.
- [2] James P. Anderson. Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co., Fort Washington, PA, 1980.
- [3] Dorothy E. Denning. An intrusion detection model. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 118-131, 1986.

- [4] B. Mukherjee, L.T. Heberlein, and K.N. Levitt. Network intrusion detection. *IEEE Network*, 8:26-41, May-June 1994.
- [5] T. Lunt *et al.* IDIS: The enhanced prototype. Technical report, SRI International, Computer Science Lab, October 1988.
- [6] D. Anderson, T. Frivold, and A. Valdes. Next-generation intrusion detection expert system (NIDES). Technical Report SRI-CSL-95-07, SRI International, Computer Science Lab, May 1995.
- [7] M. Sebring *et al.* Expert systems in intrusion detection: A case study. *Proceedings of the 11th National Computer Security Conference*, 1988.
- [8] L. T. Heberlein *et al.* A network security monitor. *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1990.
- [9] K. Jackson, D. DuBois, and C. Stallings. An expert system application for network intrusion detection. *Proceedings of the 14th Department of Energy Computer Security Group Conference*, 1991.
- [10] S. Snapp *et al.* DIDS – motivation, architecture and an early prototype. *Proceedings of COMPCON*, 1991.
- [11] D. Seely. A tour of the worm. *IEEE Trans. on Soft. Eng.*, November 1991.
- [12] Computer Security Research Group. *The Design of GrIDS: A Graph-Based Intrusion Detection System*. Technical report, UC Davis Department of Computer Science, Davis, California, in preparation.
- [13] Steven McCanne, B. Jacobsen, and Craig Leres. Tcpdump. <ftp://ftp.ee.lbl.gov>.

ATTACK CLASS: ADDRESS SPOOFING

L. Todd Heberlein

Net Squared
4324 Vista Way
Davis, CA 95616

Matt Bishop

Department of Computer Science
University of California
Davis, CA 95616

Abstract

We present an analysis of a class of attacks we call address spoofing. Fundamentals of internetwork routing and communication are presented, followed by a discussion of the address spoofing class. The attack class is made concrete with a discussion of a well known incident. We conclude by dispelling several myths of purported security solutions including the security provided by one-time passwords.

1 Introduction

Recently we began analyzing known vulnerabilities and attacks for the purpose of modeling them. We believe a sufficiently complete model will allow us to both predict new instances of general attack classes and build generic schemes for detecting exploitations of general vulnerability classes. This paper discusses one vulnerability/attack class we call address spoofing.

Many of today's network services use host names or addresses for both identification and authentication. A system using such a service composes a message and sends the message to the service on a remote system. The service on the remote system allows or disallows the request solely on the sender's address included in the request. For example, a remote login may be allowed without formal authentication (e.g., no password is required) if that remote login is coming from a "trusted" host. Table 1 describes some of the services using the senders address for authentication. Many higher level network services (e.g., network back-ups) are built on these vulnerable services thereby inheriting or extending their risks.

Unfortunately, addresses were not designed to provide authentication, and an adversary can take advantage of this fact by forging an artificial request. This paper

explores how, why, and under what conditions an adversary can exploit services using address-based authentication. Following a discussion of the problem in the most general sense, we present a specific example of such an attack. Finally, we will conclude by answering some of the questions surrounding this problem.

2 Background Fundamentals

In order to more fully understand why and how address spoofing can be performed, we first cover some of the basics of communication and routing. These basic properties will be used to characterize an adversary's capabilities and strategies.

2.1 Connectionless vs. Connection-oriented Communication

As mentioned in the previous section, an adversary exploits the services of interest by forging a message; however, before we can define what a "message" is, we must examine some of the fundamentals of network communication.

Communication across a network falls into two broad categories: connectionless and connection-oriented communication. In connectionless communication, typically supplied by a protocol layer such as UDP, no state information about previously exchanged information is kept. If a process wants to

<u>SERVICE</u>	<u>EXPLANATION</u>
r* commands	remote login, remote shell, remote copy, etc.; host address can provide authentication by .rhosts and hosts.equiv files.
mountd	file system mounting; host address is used to allow access and access rights. Host access is usually specified in a file called something like /etc/exports.
TCP/UDP wrappers	wrappers around network services; wrappers are often used to deny access except to a few hosts to network services. IP access/restriction can be set in specific configuration files.
firewalls	IP firewalls are used to restrict access into a network to certain services and certain IP addresses. IP access/restrictions can be set in configuration files.

Table 1

send a message to another process which is already waiting, the first process simply constructs the message and gives it to the connectionless protocol layer (e.g., UDP) to deliver. Because no state information is kept, the underlying protocol being used does not guarantee that messages will arrive at their destination or even if the messages will arrive in the order that they were sent. However, this lack of state also makes connectionless protocols such as UDP very efficient and therefore desirable for many network services.

Processes requiring more robust communication, at the cost of some efficiency, use connection-oriented communication; the TCP layer provides such services. Connection-oriented communication "guarantees" that information will both arrive and arrive in order at the destination process, or if delivery could not be made, at least the sending process will be notified. Connection-oriented communication goes through three phases: connection set-up, data exchange, and connection tear-down. Under TCP, the set-up and tear-down process are performed by three way handshakes; the set-up handshake is described below.

The connection set-up is a three way handshake during which each host tells the other its beginning sequence number and acknowledges the beginning sequence number of the other host (see Fig. 1). The connection is NOT considered established until both hosts have acknowledged the other host's sequence number. Once the connection is established, the sequence numbers will be used to guarantee in-order delivery of data. In the first packet exchange in figure 1, Host A (Alice) notifies Host B (Bob) that she wants to establish a connection and provides her starting sequence number X. In the second packet exchange, Bob sends his starting sequence number, Y, and acknowledges that he has received Alice's starting number (it is incremented by one). In the final exchange, Alice acknowledges that she has received Bob's starting sequence number (once again, incrementing Y by one). At this point, the connection is established and data can be exchanged.

An important feature to note is that Bob's sequence number, Y, must be used in the third part of the handshake - Alice's second packet. If Alice is not able to demonstrate to Bob that she knows his sequence number, Bob will terminate the connection before it is fully established.

2.2 Routing

Routing, under the internet protocol suite, is almost magical. A host wanting to send a packet to a remote host somewhere else on the internetwork need only place the packet on the network, and the packet will be automatically routed through the network until it reaches its destination. Neither the sending nor receiving host need to know the underlying architecture of the internetwork (hence, we often refer to an internetwork as a cloud). What is even more interesting for our needs is that, for the most part, during a packet's travels across the internetwork, only the destination address of the packet is examined. Therefore, the source address can be anything, including a non-existent host, and the internetwork will still deliver the message.

In Figure 2, our adversary E (Eve) wants to send a message to B (Bob) pretending to be A (Alice). Fortunately for Eve, she only needs to construct the packet and place it on the internet. The cloud will properly route the packet to Bob, and he will be unable to tell that it was not Alice who sent it. Once again, this feature will be important as we describe the potential attacks.

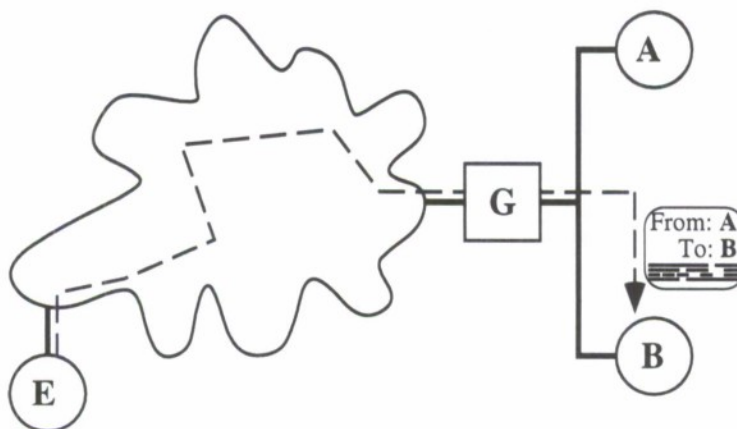


Figure 2

Connection Set-up

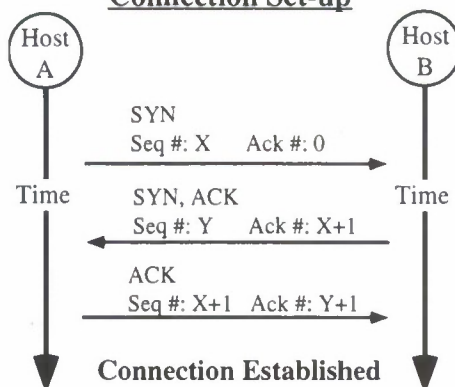


Figure 1

3 The Attack

We are now prepared to present the address spoofing attack class. In this section we will explain exactly what we consider is an attack, explain the restriction in the attack, and provide the strategy for an adversary.

3.1 Definition

Our model includes three players, Host A (Alice), Host B (Bob), and the adversary, Host E (Eve). Bob explicitly grants Alice special privileges. This granting of privileges is performed by listing Alice's name (or address) in special configuration files (e.g., .rhosts). Thus, Alice is able to get Bob to perform certain actions, actions he will not perform for just anybody, simply because she is who she says she is. Eve's goal is the following: **To get Bob to perform a specific action that he would perform for Alice but not Eve.**

3.2 Restrictions

We must concern ourselves with two major issues: (1) the placements of Alice, Bob, and Eve and (2) the nature of the communication used to get Bob to perform the desired actions.

3.2.1 Architecture

The placement of the three players can be described as the model's architecture. The most basic architecture has Alice and Bob on the same network as in figure 2. In this scenario, either Eve is also on the same network or she is outside the network. However, for the purpose of this presentation we will examine the more general architecture where Alice and Bob are on separate networks. In this scenario, Eve's location relative to Alice and Bob can be described by one of the following four categories: (1) on the same network as Bob, (2) somewhere on the path between Alice and Bob, (3) on the same network as Alice, or (4) not on either of Alice or Bob's network and not in the path of the data (see figure 3). Each of Eve's four positions will dictate different strategies used by Eve and different defensive/detection strategies used by Alice or Bob.

Please note that the simpler architecture, where Alice and Bob are on the same network, is really a special case of our more general architecture depicted in figure 3. Namely, E₁ and E₃ collapse into one case, E₄ remains as is, and E₂ is eliminated.

3.2.2 Communication Nature

Here we are concerned with how Alice and Bob normally communicate. For if Eve is to get Bob to perform some action by making him believe Alice is requesting it, Eve's communication with Bob must be indistinguishable from Alice's communication with Bob (at least from Bob's perspective). We divide communication into two broad categories we call orders and dialogues. In order communication, Alice sends a single message to Bob. Bob may reply, but we assume he has already carried out the order

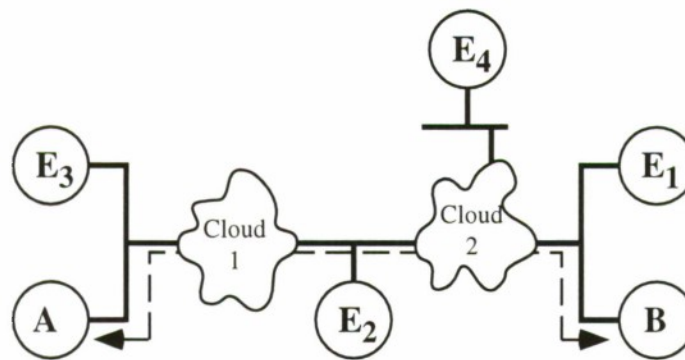


Figure 3

any request. If the dialogue does not make sense from Bob's perspective, Bob will not carry out the requested action (indeed, Bob may stop the dialogue before he even receives the request). Any communication over TCP must be considered a dialogue because as we showed earlier, several messages (packets) must be exchanged to set up a TCP/IP connection. Furthermore, Bob will be replying to Alice (not to Eve, who is pretending to be Alice). If Alice receives Bob's replies, she may tell Bob that she isn't talking to him, at which point Bob will terminate the dialogue. Eve may need to keep the dialogue going for some time, so she will need to prevent Alice from alerting Bob.

The nature of the communication, order or dialogue, used to get Bob to perform the desired action will dictate Eve's strategy.

3.3 Strategy

For Eve to complete her goal, she must achieve two main subgoals: establish a forged communication with Bob and prevent Alice from alerting Bob until it is too late. we examine each of these goals and their challenges in the following section.

For Eve to transmit a forged packet to Bob, she must simply construct the packet and place it on the network. The routing software in the network will deliver the packet for Eve. If the communication is order-based in which only a single packet is needed (e.g., a remote procedure call over UDP), then Eve has completed her communication subgoal. However, if

before replying. A popular form of order communication is the remote procedure call (RPC) over UDP.

In dialogue communication, Alice and Bob exchange several messages prior to Bob carrying out

communication is dialogue-based, Eve will need to send multiple packets to Bob, the contents of which will depend on replies that Bob makes (e.g., Bob's sequence number under TCP). If Eve is in positions E₁, E₂, or E₃, she is able to observe Bob's responses thereby allowing her to send meaningful subsequent packets to Bob. If Eve is in position E₄, she can still observe Bob's responses if she is able to modify the reply path from Bob to Alice. This can easily be done through source routing in IP networks. Modifying router settings are also an option to Eve. Finally, even if Eve is in position E₄ and is unable to direct Bob's traffic to Alice through Eve's own network, if Eve can predict Bob's responses (e.g., what Bob's sequence number will be), she can still carry on the communication with Bob. Predicting sequence numbers is discussed in [Morris 85] and [Bellovin 89] and was used in the recently IP spoofing attack.

Eve's second major goal is to prevent Alice from interfering with the attack. Eve can achieve this goal in many ways; we will discuss three: (1) prevent the packets from reaching Alice (or Alice's packets from reaching Bob), (2) take away Alice's ability to respond, and (3) complete the communication before Alice's alerts can reach Bob. The first approach requires Eve to modify the routing behavior of the network. If Eve is a node in the routing path (e.g., she is a router or has used source routing to make the route flow through her), she simply doesn't forward the packets to Alice. Even if Eve is not on the path between Alice and Bob, she could modify the routing information in one of the routers in the path to misroute Alice's packets. Eve could also wait for the internetwork between Alice and Bob to fail and launch her attack then.

The second approach, taking away Alice's ability to respond, can be much simpler for Eve to implement. Eve can (1) cause Alice to crash (not terribly difficult), (2) wait for Alice to go down for other reasons (e.g., maintenance), or (3) block the TCP/IP portion of Alice's operating system so that it cannot process Bob's packets. This latter approach, perhaps the most graceful, was used in the recently publicized IP

spoofing attack and originally detailed in [Morris 85].

The third approach, completing the communication (at which time Bob has completed the action) before Alice can alert Bob, is trivial in the order-based communication (e.g., RPC). Bob will have completed any operation prior to sending any messages to Alice; therefore, by the time Alice is aware that something is wrong, she is too late. For dialogue-based communication, the solution is more difficult, because Bob will be sending data to Alice before Bob completes the requested operation. However, if the communication between Eve and Bob is much faster than between Alice and Bob, Eve could complete the attack in time.

3.4 Attack Summary

For Eve to achieve her goal of getting Bob to perform an action for Eve when he thinks he is doing it for Alice, Eve must (1) get the forged message to Bob, (2) if necessary carry on a dialogue with Bob, and (3) prevent Alice from interfering with the communication. Internetwork routing will usually take care of the first subgoal for Eve. The last two goals may be achieved in a number of ways; our suggested approaches were by no means complete.

4 An Example Attack

Having mapped out a general plan for Eve to exploit access control which is based on IP addresses or names, we now examine a particular instance of such an attack. The attack, launched at the end of 1994 against Tsutomu Shimamura's machine, has gained the attention of the popular press, the usenet, and CERT. The attack can be mapped directly onto the our general model. Furthermore, the attack was explicitly described in [Morris 85].

This particular attack involved a server (Alice) and an X-client (Bob) (see figure 4). Eve was in position E₄. That is, she was unable to observe the messages passing between Alice and Bob.

Step 1: "Wedge" a portion of Alice's OS such that it cannot process Bob's replies.

Players	E	adversary
	A	server
	B	X-client
Steps	1	Prevent Alice From Responding
	2	Probe for sequence number prediction
	3	Forge communication

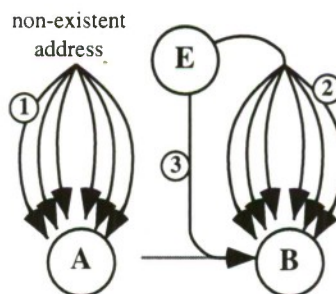


Figure 4

This is performed by sending multiple connection requests to the rlogin port (port 513) from a non-existent host. Alice responds to each request (the second part of the TCP handshake), but since the originating host does not exist, the third part of the handshake never comes. Alice is left with several partially opened connections, each filling up space in her internal data structures. Alice is only able to support up to eight of these partial connections before internal tables fill up and she stops responding to new packets to port 513.

Please note that had Eve listed her own IP address in the forged, artificial requests, her own TCP/IP software would have sent a reset command, RST, following Alice's reply. The RST would have freed Alice's data structures. Therefore, Eve had to use an artificial address as the sender of the requests—one that would never reply to Alice's responses.

Step 2: Predict what Bob's sequence number will be. Eve sent 20 connection requests to Bob's remote shell server; the starting sequence number for each connection request incremented by a predictable value of 128,000. Eve most likely used an address of a legitimate host for the connection requests. Thus, when Bob replies with the second part of the handshake, the OS, which did not actually make the initial request, generates a reset message (RST). This prevents Bob's operating system from wedging in the same way Alice's did in Step 1. The address of the forged requests may either be Eve's own or the address of another host. If the address is that of another host, Eve must be able to observe the path between Bob and the other

host (in order to observe the sequence numbers).

Step 3: Have a dialogue with Bob pretending to be Alice (which is still in a confused state). In this particular case, the dialogue was a TCP/IP connection to the remote shell server. Although Eve could not see Bob's replies, she accurately predicted that Bob's starting sequence number would be 2,024,384,000—exactly 128,000 more than the last requested shell connection in step 2. Eve's requested action: place a "+" in the /.rhosts file (a shell command such as "echo '+' >> /.rhosts" was sent). Bob, believing the connection was from Alice, carried out the request.

At this stage, the goal we set forth for Eve has been completed. She was able to get Bob (the X-client) to perform an action (place a "+" in /.rhosts) for her; something only Alice could legitimately do. Following this attack, the adversary easily logged into Bob via rlogin. In fact, at this point anyone from anywhere could rlogin to Bob.

5 Popular questions

Couldn't this attack be simply stopped by configuring routers (or firewalls) to not forward an obviously forged packets? This is true in limited circumstances. For example, if the gateway G in figure 2 did not forward the packet which already states that it is from A onto A's network, then the forgery in figure 2 could not take place. However, this is only a partial solution. If hosts A and B (Alice and Bob) are not on the same network already, this approach cannot work. Furthermore, even if Alice and Bob are on the same

network, this cannot prevent attacks coming from Eve if she is on the same network already. In short, this solution is limited in scope. A solution should not be dependent on the network architecture.

In general, we define the Point of Convergence as the point where the path from Alice to Bob and Eve to Bob become the same path (see Figure 5). A monitor or gateway placed between the Point of Convergence and Bob would not be able to detect or stop masquerading traffic from Eve.

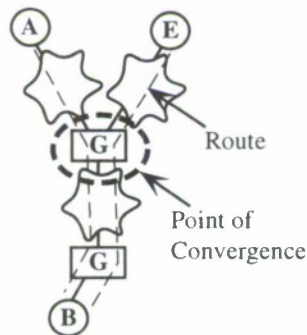


Figure 5

Couldn't we require all "trusted" hosts to belong to the same physical network (that is, no traffic passes through a gateway) and use lower level addresses such as the ethernet address? No. While it is widely believed that the ethernet address is a property of a host's ethernet hardware and is therefore unchangeable, we have demonstrated the creation of packets which include a forged ethernet source address. These packets are indistinguishable from legitimate packets. We have reason to believe that this is true with other network media as well.

Couldn't we simply write a more secure algorithm for choosing initial sequence numbers? If by "secure" you mean a less predictable starting sequence number, the answer is again true, but only in limited circumstances. This would work if, as in the case in figure 3, the adversary Eve is in position E₄ and unable to alter routing information to get the traffic to flow through her. However, if Eve is in positions E₁, E₂, or E₃ or if Eve is in position E₄ and can use source routing or other means to alter

routing, a more random initial sequence number would still not work. Eve is still able to observe Bob's sequence number.

What other extensions to this attack might exist? While numerous possibilities exist, perhaps the one which concerns us the most is the placing of a forged request into an already existing TCP/IP connection. This approach is generally referred to as session hijacking, and existing programs, including commercial ones, already have this capability. In such an attack, even password protected services are vulnerable since every packet following a correct sign-on is trusted. Indeed, we have even demonstrated this attack against systems with one-time passwords.

6 Summary

We have described a class of attacks we call address spoofing. The reason this class exists rests squarely on the fact that systems and application developers have chosen to use a property which was not designed to provide security, namely the sender's network address, as a means of authentication. We have outlined where and how this vulnerability can be exploited, and we described a real instance of such an attack. Finally, we hope to have convinced you that the solution is not with "fixing" parts of the protocols (addresses and sequence numbers) which are not broken, but with getting systems and application developers to build their security on properties developed with the purpose of providing security in the first place.

References

- [Bellovin 89] Bellovin, S., "Security Problems in the TCP/IP Protocol Suite," *Computer Communication Review*, Vol. 19, No. 2, pp. 32-48, April 1989.
- [Morris 85] Morris, R.T., "A Weakness in the 4.2BSD Unix TCP/IP Software," *Computing Science Technical Report No. 117*, AT&T Bell Laboratories, Murray Hill, New Jersey.

Generic Model Interpretations: POSIX.1 and SQL

D. Elliott Bell*

Mitretek Systems
7525 Colshire Drive
McLean VA 22102

Abstract

An improvement to the traditional process of model interpretation is described. The improvement applies to trusted systems that conform to industry standards that are conducive to generic model interpretation. Generic model interpretation results for POSIX.1 and SQL are presented.

KEYWORDS: Security model, model interpretation, POSIX, SQL, TCSEC

Introduction

The modeling requirements for Division B of the *Trusted Computer System Evaluation Criteria (TCSEC)* [TCSEC85] derive from the very earliest experiences in conceiving and producing trusted computer systems. The Anderson report [AND72] called for a "conceptual design, that is a mathematical model" of a secure computer system as part of the plan to realize the "Reference Monitor Concept" in an implementation. Efforts during the 1970's and early 1980's (see for example [WALT74], [BLP73], [LPB73], [BELL73], [BLP75], [GOME82], [GOME84]) included conceptual design tools in the form of mathematical models for use in assessing the (defined) security of systems and products of interest.

Work before (and system and product evaluation since) the publication of the *TCSEC* has been very consistent in terms of the elements and the correspondences that are required in a

model interpretation of a specific implementation. In *TCSEC* terms, what is required is a Formal Security Policy (FSP), a Formal Security Policy Model (FSPM), a Descriptive Top-Level Specification (DTLS), and connections between them.

The required connection between the model and the policy is that it be a model *of* the policy in question so that modeling results will allow policy-relevant statements to be made about the system at issue. The model itself must also be shown to be sound and free from defect.

There must also be a (descriptive top-level) specification of the system. This abstraction of the system at its interface must be shown to *correspond* to the model. The notion is that the model results — usually of the form "this transition preserves the important notions of 'security'" — are inherited by the system at the level of the specification if the individual system calls (or gates or entry points) exhibit the same behavior as model transitions that have been analyzed and found "secure".

In terms of the "design assurance chain" shown in Figure 1, there must exist documentation for



Figure 1: Design Assurance Chain

POLICY, MODEL, and DTLS, there must be arguments that the correspondences indicated by the arrows between them are available, and the ar-

*Work performed under contracts MDA904-93-C-C003 and MDA904-96-C-0619.

guments for correspondence must be *convincing*. The focus of this paper is Model Interpretation (MI), shown as the right-most arrow of Figure 1. All five conditions (policy, model, DTLs, and two arrows) are termed the "B2 checks".

Product evaluation under the *TCSEC* thus requires (a) a generic (formal security) policy; (b) a (formal security policy) model of that policy, together with results that assure its soundness and freedom from error; (c) a (descriptive top-level) specification of the TCB interface; and (d) a correspondence between the DTLs and the model, the model "interpretation".

Satisfaction of the B2 checks for a proprietary computing system builds directly on that system's idiosyncratic characteristics at the boundary of its Trusted Computing Base (TCB). Thus while general modeling resources can be used, the step of matching the system to the modeling entities is by nature specific to the system.

The trend in trusted products towards compliance with industry standards — especially the Portable Operating System Interface (POSIX) and the Standard Query Language (SQL) — makes possible a revision of the traditional procedure for satisfying the B2 checks. Rather than matching every system design directly to a model, one can produce a generic model interpretation for each industry standard, thereafter using the results to simplify and reduce the task of model interpretation for conforming implementations.

In terms of the generic "design assurance chain" in Figure 2, the Generic Model Interpretation



Figure 2: Generic Design Assurance Chain

(GMI) from the standard to the model will be available at the time that the product-specific model interpretation begins. Since a conforming implementation will match the specification of the standard by definition, a model interpretation task will be limited to any minor differences between the standard and the product. The production and review of the resulting model interpretation will be easier tasks, smaller tasks, and tasks that require substantially less time to perform.

This paper presents the results of a recent effort to create generic model interpretations for both

POSIX and SQL (see [POSIX.1] and [LEFF91], respectively).

The next two sections describe the processes of producing and using a generic model interpretation. The following two sections describe the results of applying this process to POSIX.1 and SQL. The final section summarizes the work.

Producing a Generic Model Interpretation

The model interpretation portion of the B2 checks consists of matching part of the system's TCB interface to the security model being used. The results can be viewed conceptually as three tables. The first table (Figure 3) identifies which of the

Name	Interface?	Reason
module-001	no	...
module-002	yes	...
module-003	yes	...
⋮	⋮	⋮
module-998	no	...
module-999	yes	...

Figure 3: TCB Interface Table

TCB modules are visible at the TCB interface. The second table (Figure 4) identifies which of

Name	Model?	Reason
module-002	no	...
module-003	yes	...
module-017	no	...
⋮	⋮	⋮
module-952	no	...
module-999	yes	...

Figure 4: Modules to Model

the TCB-interface modules should be interpreted in modeling terms. The last table (Figure 5) lists the model rules (or transitions) that correspond to the TCB call.

The construction and justification of these three tables constitutes the bulk of the task of "model interpretation". In an ideal situation, the system or product would be stable and fully documented. There would also be resources available for answering questions and providing clarification. The

Name	Rule
module-003	ρ_3
module-042	ρ_{12}, ρ_5
module-116	ρ_4
\vdots	\vdots
module-666	$\rho_{12}, \rho_7, \rho_{16}$
module-999	ρ_7

Figure 5: Correspondence to Model

effort of constructing the first table would begin with a list of all the modules that constitute the TCB. Review of each of the modules would determine whether or not it is part of the TCB interface, and whether it is "visible" at the TCB interface. The subset of the modules identified as "at the interface" in the construction of the TCB-interface table would be the starting point for the modules-to-model table. The determination of whether a module should be modeled is more complex. It is dependent on such things as which system abstractions are exported by the TCB, which modules are available to ordinary users (rather than just to specially-privileged users), whether the module's action is part of access-control-policy mediation, and whether the module's action is a null action in a modeling context. Since the process of determining *whether* a module should be modeled is tightly coupled with the assignment of a corresponding model rule (or rules), the construction of the model-correspondence table will usually proceed in parallel with the construction of the modules-to-model table.

In less ideal situations (such as system model interpretations and product model interpretations while the system is still in flux), the same tables must be constructed, but the available materials are less complete. One must therefore proceed cautiously, realizing that many conclusions will have to be conditional and may have to be revised when more information becomes available.

Analysis of a generic industry standard entails a double-pronged effort. The first effort is the review of all the system calls according to their type. The set of types is developed inductively as the analysis proceeds. The second analytical approach is to assess the system calls with regard to their availability to different classes of users and to their relation to the enumerated policies that are of interest. This analysis is referred to as the "SAP"

analysis.¹ A set of guidelines for SAP analysis is provided in the annex.

Consideration of both the types and SAP results will allow a determination of which TCB modules are the proper ones for generic model interpretation. The final step is identifying the correspondences between system calls and model rules to complete the construction of the generic model interpretation.

Using a Generic Model Interpretation

A generic model interpretation provides a starting point for the review of a model interpretation of a conforming implementation in the form of a Correspondence-to-Model table. A completed model interpretation for a specific implementation will also include a Correspondence-to-Model table. A comparison of those two tables will be the first step in evaluating the specific model interpretation.

Name	Rule
mod- n_1	ρ_{n_1}
mod- n_2	ρ_{n_2}
mod- n_3	ρ_{n_3}
\vdots	\vdots
mod- n_{k-1}	$\rho_{n_{k-1}}$
mod- n_k	ρ_k

GMI

Name	Rule
mod- m_1	ρ_{m_1}
mod- m_2	ρ_{m_2}
mod- m_3	ρ_{m_3}
\vdots	\vdots
mod- m_{k-1}	$\rho_{m_{k-1}}$
mod- m_k	ρ_{m_k}

MI

Figure 6: Comparison of Two Correspondences

The two model-correspondence tables have four modes of comparison:

1. an entry in the specific table matches an entry on the generic table exactly in both the "name" and "rules" value;
2. an entry in the specific table matches the "name" value, but the corresponding rules are not the same;

¹The designation "SAP" derives from the codes used during the analysis: "S" refers to system calls that are available to a Standard user. "A" refers to system calls that include some extra effect for users with Appropriate privilege. "P" refers to system calls that relate to the enumerated Policies.

3. an entry in the specific table does not match any entry's "name" value in the generic table; and
4. an entry in the generic table does not match any entry's "name" value in the specific table.²

The generic model interpretation results apply to the specific implementation in the first case. In the other cases, the anomaly has to be resolved. The question is not "what mistake has been made in the specific model interpretation?" but "why is there this difference?" The reasons could be

- an error in generic model interpretation;
- an error in specific model interpretation; or
- differences between the specific case and the generic cases justify different results.

An initial comparison of the specific model interpretation table with the generic table has the beneficial effect of focusing attention on that portion of the modeling interpretation that is most in need of consideration.

The use of the generic interpretation is similar to a situation where one is attempting to construct a model interpretation during the completion of the implementation phase. In this case, the task is the construction of the model-correspondence table. The generic model-correspondence table provides version-zero of the required table. Meticulous review of the specific implementation in comparison to the generic model interpretation will allow the identification of both those rows that match the specific conforming implementation exactly and those system calls that need direct analysis and treatment.

GMI for POSIX.1

POSIX.1 is an especially lucrative target for generic model interpretation. Not only are there many trusted operating systems pledged to POSIX.1 compliance, but also there is voluminous open literature concerning Unix and POSIX.1 (see, for example, [POSIX.1], [USL92a], [USL92b]).

²There is the possibility, of course, that both the generic and the specific model interpretations are erroneous, but that their errors compensate, making the rows match each other. It is assumed that independent commission of such mistakes by model-interpreters is rare enough to be disregarded.

Types	Code
Access	A
A/policy control	A/C
A/post-mediation access	A/A
A/other factors	A/O
A/O/groups	A/O/
A/O/owner	A/O/O
A/O/pathname resolution	A/O/P
A/O/attributes	A/O/A
A/O/id's	A/O/I
Process	P
P/address space	P/A
P/memory	P/M
Job Control	J
J/signals	J/S
J/locks	J/L
J/pipes	J/P
Files	F
F/descriptors	F/D
F/link	F/L
IPC	I
I/msgs	I/M
I/sEmaphores	I/E
I/sHared memory	I/H
Resources	R
R/devices	R/D
R/D/STREAMS	R/D/S
R/affinity	R/A
R/file systems	R/F
Operations	O
O/networking	O/N
O/stat	O/S
O/auDit	O/D
O/auThentication	O/T
Maglc	M
vfun's	V

Figure 7: "Types" of POSIX System Calls

System Call	Corresponding Rules
exit	release-access
accept	get-access
chmod	grant/rescind-access
close	release-access
connect	get-access
creat	create-object
fcntl	create/delete-object
mknod	create-objects
mkdir	create-objects
mount	create-objects
sem_release_id	release-access, delete-object
setoacl	grant/rescind-access
setomac	change-object-level
dup	get-access
dup2	get-access
fchmod	grant/rescind-access
mkdir	create-object
mount	create-objects
msgctl	release-access, delete-object
msgget	create-object, get-access
open	get-access
rmdir	delete-object
semctl	release-access, delete-object
semget	create-object, get-access
shmat	get-access
shmctl	release-access, delete-object
shmdt	release-access
shmget	create-object, get-access
socket	create-object
socketpair	create-object
symlink	create-object
umount	delete-subtree
mknod	create-object

Figure 8: POSIX.1 Correspondence to Model

The situation of treating POSIX.1 generically is like the non-ideal situations above, but there is no expectation that more information will ever be available. That is, the details of a specific conforming implementation will never be available until that model interpretation is undertaken. Because of this intrinsic conditionality, the generic modeling interpretation will adopt a variation of the ideal approach described above.

In the context of a specific system, the full list of TCB modules will be a superset of the required POSIX.1 system calls. In the general case, therefore, the table of TCB modules constructed will necessarily be conditional and incomplete. The best that can be done is to construct a best-efforts modules-to-model table, a list of those POSIX.1 system calls that should usually be interpreted in modeling terms. From that conditional table, a corresponding conditional model-correspondence table can be constructed.

In sum, the generic-model-interpretation plan for POSIX.1 is to construct a table of POSIX system calls that will normally be modeled, together with corresponding model rules.

Analysis of POSIX.1 identified the types shown in Figure 7.³ The post-mediation access subtype was included with the mandatory types for model interpretation. The correspondence to model rules is shown in Figure 8.

GMI for SQL

SQL is also a lucrative target for model interpretation, but the wide variety in possibilities for Reference-Monitor-protected “objects” makes a comprehensive generic model interpretation a larger task than for POSIX.1. The results included in [MS96b] and summarized here constitute an initial step in generic model interpretation for SQL. The correspondences derived were produced by limiting the scope of attention to databases, tables, rows, view definitions, and columns.⁴

As is the case for POSIX.1, there is substantial open literature concerning SQL (see, for example, [BED93], [LEFF91]).

Analysis of SQL identified the types shown in Figure 9. The post-mediation access subtype was

³It is important to note that the types used for this analysis were defined subjectively and are in no way unique or necessary. Moreover, the types are not disjoint. In fact, some of the types were explicitly noted as overlapping with other types.

⁴Other object-candidates are indexes, constraints, and stored procedures.

Types	Code
Access	A
A/policy control	A/C
A/post-mediation access	A/A
A/other factors	A/O
SQL	SQL
Admin & Support	A&S
security	S
S/Grant,Revoke	S/GR
S/Audit Support	S/A
S/Tier 2 Audit Layer	S/A2
S/M.A.C.	S/M
S/D.A.C.	S/D
S/Discrete Privilege	S/P
Data Integrity	I
I/concurrency	I/C
I/transaction	I/T
I/recovery	I/R
I/mirroring	I/M
I/archiving	I/A
Object Management	O
O/DB Management	O/D
O/System Catalog	O/SC
O/Table	O/T
O/Row Data	O/R
O/Index	O/I
O/Constraint	O/C
O/View	O/V
O/Synonym	O/Y
O/Statistics	O/S
O/In-Core Dictionary	O/Lex
O/Stored Procedures	O/SP
Data	D
D/DB Services	D/D
D/B-tree	D/Bt
D/Row	D/R
Resource Management	R
R/DB space & chunk	R/D
R/Page	R/P
R/Page & slot	R/PS
R/Shared memory	R/SM
R/Buffer	R/B
R/Header	R/H
Magic	M
vfun's	V

Figure 9: "Types" of SQL Commands

SQL Call	Corresponding Rules
Change Process Label	change-object-level
Create Database	create-object
Drop Database	delete-object
Open-Lock Database	get-access
Close-Unlock Database	release-access
Create System Catalog	create-object
Open System Catalog Table	get-access
Drop System Catalog	delete-object
Read System Catalog	null transition
Write System Catalogs	null transition
Create Table	create-object
Drop Table	delete-object
Alter Table	null transition
Fast Alter Table	null transition
Open-Lock Table	get-access
Close Table	release-access
Insert Row	create-object
Delete Row	delete-object
Select Row	null transition
Update Row	null transition
Create View	create-object
Drop View	delete-object
Create Database Entry	create-object
Open Database Entry	get-access
Drop Database Entry	delete-object
Close Database	release-access
Grant Privilege	give-access
Revoke Privilege	rescind-access
Modify Database Label	change-object-level
Modify Table Sensitivity Label	change-object-level
Modify Row Sensitivity Label	change-object-level
Grant Table Level Privilege	give-access
Revoke Table Level Privilege	rescind-access
Grant Database Level Privilege	give-access
Grant Table Level Privilege	give-access
Revoke Database Level Privilege	rescind-access
Revoke Table Level Privilege	rescind-access

Figure 10: SQL Correspondence to Model

included with the mandatory types for model interpretation. The correspondence to model rules is shown in Figure 10.

Summary

The benefits of producing generic model interpretations as a tool to facilitate system and product evaluation are many. The work reported herein has both developed the theory of generic model interpretations and begun the task of providing generic model interpretations for POSIX.1 and SQL. The POSIX.1 results are very comprehensive, a result both of the more extensive information available on POSIX and of the narrower scope available for vendors producing a POSIX-compliant product that also meets B2 or above requirements from the *TCSEC*. The SQL results are also comprehensive for the object candidates addressed, but further work will be required to complete the consideration of all possible object candidates.

To the benefits provided by industry standards can now be added the ability to facilitate model interpretation preparation and review through the use of generic model interpretations.

References

- [AND72] J. P. Anderson, "Computer Security Technology Planning Study", ESD-TR-73-51, Vol. I, AD-758 206, ESD/AFSC, Hanscom AFB, MA, October 1972.
- [BLP73] D. E. Bell and L. J. La Padula, "Secure Computer Systems: Mathematical Foundations", MTR-2547, Vol. I, The MITRE Corporation, Bedford, MA, 1 March 1973. (ESD-TR-73-278-I)
- [BELL73] D. E. Bell, "Secure Computer Systems: A Refinement of the Mathematical Model", MTR-2547, Vol. III, The MITRE Corporation, Bedford, MA, December 1973. (ESD-TR-73-278-III)
- [BLP75] D. E. Bell and L. J. La Padula, "Secure Computer Systems: Unified Exposition and Multics Interpretation", MTR-2997, The MITRE Corporation, Bedford, MA, July 1975. (ESD-TR-75-306)
- [BL86] D. E. Bell, "Secure Computer Systems: A Network Interpretation", Proc., 2nd Aerospace Conference, McLean, VA, 1986, 32-39.
- [BL86] D. E. Bell, "Trusted Xenix Interpretation: Phase 1", Proc. 13th NCSC, Washington, DC, 1990, 333-339.
- [BIBA77] K. Biba, "Integrity Considerations for Secure Computer Systems", The MITRE Corporation, Bedford, MA, April 1977.
- [BED93] J. S. Bowman, S. L. Emerson, M. Darnovsky, , *The Practical SQL Handbook*, 2nd ed. (Addison-Wesley: Reading, MA, 1993)
- [GOME82] J. A. Goguen and J. Meseguer, "Security Policies and Security Models", *Proc. 1982 IEEE Symp. on Security and Privacy*, Oakland, CA, April 26-28, 1982, 11-20.
- [GOME84] J. A. Goguen and J. Meseguer, "Unwinding and Inference Control",

Proc. 1984 IEEE Symp. on Security and Privacy, Oakland, CA, April 29-May 2, 1984, 75-86.

System", *AFIPS Conf. Proc.* 35
FJCC 1969, 27-38.

- [LPB73] L. J. La Padula and D. Elliott Bell, "Secure Computer Systems: A Mathematical Model", MTR-2547, Vol. II, The MITRE Corporation, Bedford, MA, 31 May 1973. (ESD-TR-73-278-II)
- [LEFF91] *Using INFORMIX-SQL*, 2nd ed. (Addison-Wesley: Reading, MA, 1991)
- [MS96a] "Generic Model Interpretation of POSIX.1", Mitretek Systems, McLean, VA, to appear.
- [MS96b] "Generic Model Interpretation of SQL", Mitretek Systems, McLean, VA, to appear.
- [POSIX.1] Information Technology - Portable Operating System Interface (POSIX) - Part 1: System Application Program Interface (API) [C Language]. ISO/IEC 9945-1:1990, September, 1990.
- [TCSEC85] *Department of Defense Trusted Computer System Evaluation Criteria*, DoD 5200.28-STD, December 1985.
- [TDI91] *Trusted Database Management Interpretation of the Trusted Computer System Evaluation Criteria*, NCSC-TG-021, Version 1, April 1991.
- [USL92a] *Programming with Unix System Calls: UNIX SVR4.2*. (Prentice-Hall: Englewood Cliffs, NJ, 1992)
- [USL92b] *Operating System API Reference UNIX SVR4.2*. (Prentice-Hall: Englewood Cliffs, NJ, 1992)
- [WALT74] K. G. Walter, *et al.* "Primitive Models for Computer Security", ESD-TR-74-1147, Electronic Systems Division (MCIT), Air Force Systems Command, Hanscom AFM, Bedford, MA, January, 1974.
- [WEIS69] C. Weissman, "Security Controls in the ADEPT-50 Time-Sharing

SAP Analysis Guidelines

1. One can determine for each system security call whether a standard user (that is, one running without "appropriate privilege") can exercise the call and whether an administrative user (that is, one running with "appropriate privilege") can exercise the call with additional options or different effect.
2. The security policies that will be addressed in the model and modeling interpretation can be specified precisely.
3. For this exercise, the specified security policies are simple-security, discretionary-security, and information flow in the exact form of the *-property.
4. One can then determine whether each system security call relates to the enforcement of the specified policy (really, "policies").
5. System Calls that a standard user can invoke (indicated by S) and that relate to enforcement of the specified policies (indicated by P) will definitely be modeled.
6. System Security Calls that an administrative user can invoke (indicated by A) and that relate to enforcement of the specified policies (P) are options for modeling.
7. The decision to model an AP system security call will be addressed on a case-by-case basis. Back-of-the-envelope rationale for an inclusion or exclusion will be generated in the consideration of each AP system security call.
8. Calls that do not relate to the enforcement of the specified security policies and calls whose effects are invisible in the model's state will not be modeled.

Generic Model Interpretations: POSIX.1 and SQL

D. Elliott Bell

`bell@mitretek.org`

Mitretek Systems

7525 Colshire Drive

McLean VA 22102-7400

001680 22/07

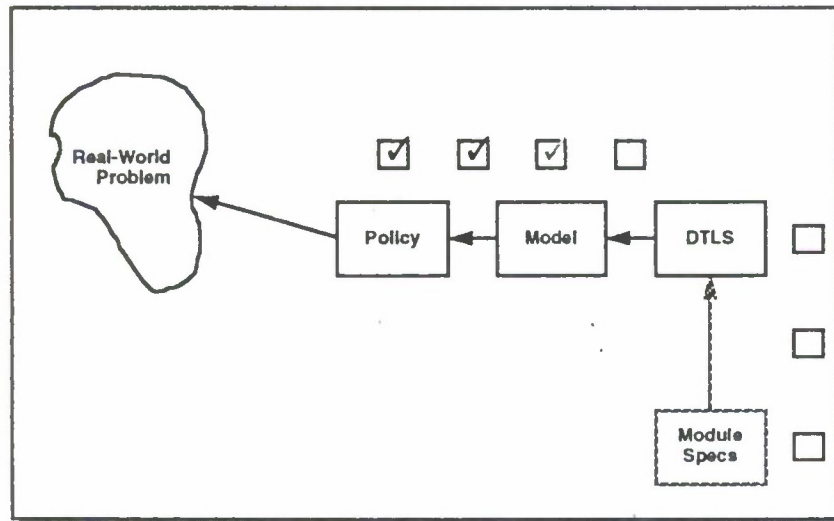
2

Outline

- Generalizing “Model Interpretation”
- Producing a Generic Model Interpretation
- Using a Generic Model Interpretation
- GMI for POSIX.1
- GMI for SQL

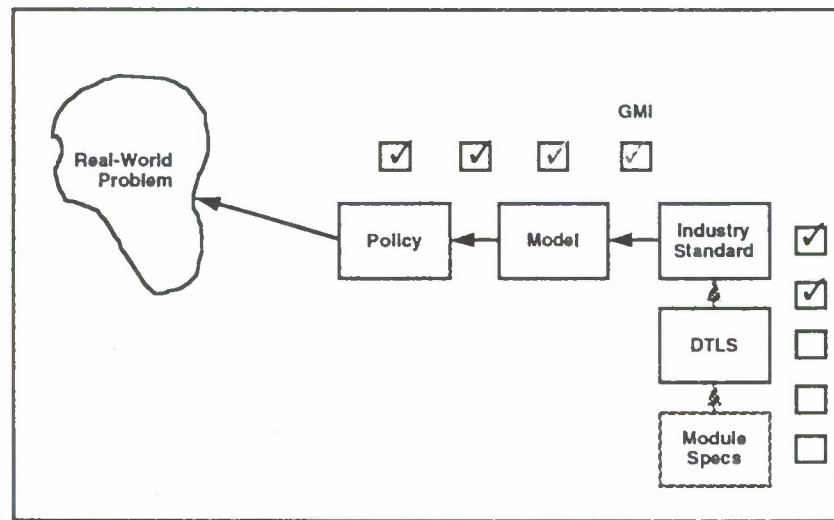
MITRETEK
SYSTEMS

Model Interpretation: Design Assurance Chain



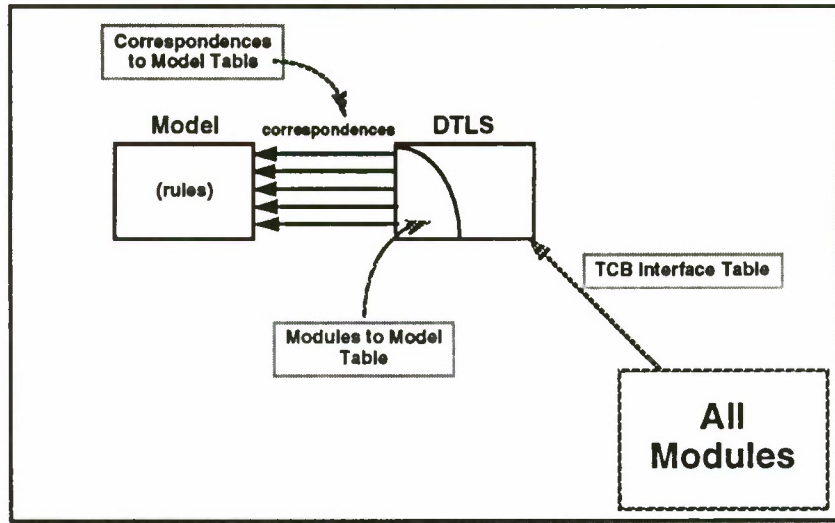
MITRETEK
SYSTEMS

Generic Model Interpretation: Design Assurance Chain



MITRETEK
SYSTEMS

Producing a Generic Model Interpretation



MITRETEK
SYSTEMS

Using a Generic Model Interpretation

<u>Name</u>	<u>Rule</u>
module _{n1}	ρ_{n1}
module _{n2}	ρ_{n2}
module _{n3}	ρ_{n3}
.	.
.	.
.	.
module _{nk}	ρ_{nk}

GMI

<u>Name</u>	<u>Rule</u>
module _{m1}	ρ_{m1}
module _{m2}	ρ_{m2}
module _{m3}	ρ_{m3}
.	.
.	.
.	.
module _{mk}	ρ_{mk}

MI

MITRETEK
SYSTEMS

THE PRIVILEGE CONTROL TABLE TOOLKIT: AN IMPLEMENTATION OF THE SYSTEM BUILD APPROACH

Thomas R. Woodall
Hughes Aircraft Company
2000 E. Imperial Highway
El Segundo, CA 90245
310-334-7603
FAX: 310-334-1242
twoodall@msmail4.hac.com

Roberta Gotfried
Hughes Aircraft Company
2000 E. Imperial Highway
El Segundo, CA 90245
310-334-7655
FAX: 310-334-1242
rgotfried@msmail4.hac.com

1. Abstract

This paper describes the Privilege Control Table (PCT) Toolkit as developed for military real-time embedded avionics systems. This tool is the evolutionary result of research and development of the trusted 'System Build' concept as originally described in [SYSBUILD] and refined in [SYSBLL]. This paper describes what the tool does and how it is used.

Keywords: System Build, Real-time, Multilevel Security

2. Introduction

The shrinking DoD budget is driving military weapon system architects to design affordable systems with greater lethality, survivability, and flexibility requirements. This is particularly true of air vehicle weapon systems, where funding limits the number of systems and aircraft that can be developed and deployed. For avionics systems, this is one of many factors that has led to integrated avionics architectures. Integrated avionics systems cost less than comparable federated systems and have the advantage of being able to provide information with increased accuracy and understandability to the pilot. To further enhance lethality, real-time sharing of information among coordinating assets is now required. In effect, the avionics system of one aircraft is one node of a distributed C4I network. This leads to an avionics system that is processing an extremely wide range of information, from unclassified weather data to highly classified and compartmentalized

information from off-board assets. This results in the need for a reliable, high performance, multilevel secure, embedded avionics system.

In past avionics systems, the hard scheduling deadlines provided difficult design challenges. Now, these real-time systems must additionally meet stringent security requirements. In order to mitigate the security impact on real-time performance, the 'System Build' concept was developed as described in [SYSBUILD] and [SYSBLL].

This paper focuses on the PCT Toolkit, a System Build tool. Section 3 briefly summarizes the System Build approach and its motivations and enabling factors; it also gives a system overview of the PCT Toolkit. Section 4 describes in detail what functions the PCT Toolkit performs, Section 5 discusses outstanding security issues associated with the Toolkit, and Section 6 summarizes the advantages and disadvantages of the PCT Toolkit and identifies areas of further work.

3. The System Build Approach

The System Build approach was developed as a result of merging two seemingly conflicting requirements: hard real-time performance and multilevel security. Few systems have had to meet both requirements simultaneously, particularly in an embedded avionics environment. Results from research in real-time and secure systems provided inadequate solutions to meeting requirements for systems to be built (target systems).

3.1 System Characteristics

Integrated avionics systems comprise a closely connected network of heterogeneous processing nodes connected to a set of sensors. Real-time characteristics of these systems are described by specific performance requirements in terms of throughput, I/O and data bandwidth, as well as interrupt latency constraints in microseconds. Such systems use preemptive priority-based process scheduling.

Mission success depends on the ability of the system to process sensor data within time constraints as well as to adapt rapidly to changes in the environment.

It is essential that security mechanisms not adversely impact the ability of the system to meet timelines, or respond to external events in a timely manner. Even small impacts in response times can have an adverse affect on the ability of the system to perform functions such as detecting and tracking targets and responding to events. Security must be achieved without sacrificing real-time performance.

3.2 Subjects/Objects Known *A Priori*

A key enabler of the System Build approach is that all subjects and objects are known *a priori*. This is a result of the embedded nature of the avionics environment. The *a priori* method is also applicable to a wide range of systems that have a fixed set of application configurations over a period of time.

The System Build approach allows access mediation at the time the software configuration is “built.” All interactions of security subjects and objects are known and may be verified for consistency with the system security policy before the software is loaded into the target system. The build-time tool that performs this function is the PCT Toolkit.

3.3 Built for a Specific Configuration

The set of system subjects and objects for a specific software load is given a unique identifier and is known as a System Build Configuration. In today’s avionics systems, a software load may not change for several months. Furthermore, the hardware configuration associated with the software will also not change for the lifetime of the

software load. This fixed combination of hardware and software components is also called the System Build Configuration. See Figure 3-1.

The inputs to the PCT Toolkit are a subset of the information that defines a System Build Configuration. Other information includes system attributes, such as the hardware configuration, that are held constant for all capabilities.

The definition of a System Build Configuration allows for a range of System Build Capabilities to be supported. For example, a System Build Capability might define the possible combinations of sensor data processing for a specific set of available processing assets. The specific set of capabilities used in a system load depends on factors which are transparent to the Toolkit. However, each specific capability must be defined by the system designer so that the Toolkit can determine what set of subjects may be allowed to run simultaneously.

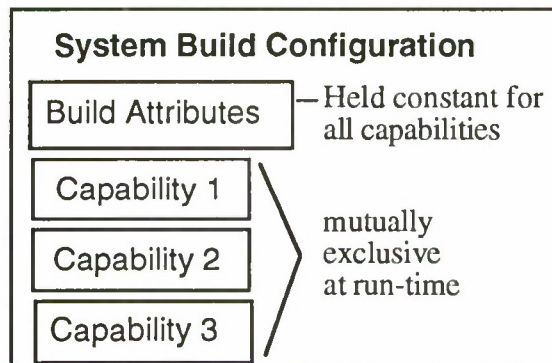


Figure 3-1: A System Build Configuration

3.4 System Engineering Methods

Embedded avionics systems are developed using rigorous engineering methods. System complexity and size, in conjunction with safety and mission criticality factors, have led to a methodology that results in extensive design analysis and documentation. Tool support for system and software engineering methods results in on-line “databases” and “documentation” that can be leveraged in implementing System Build. Application/software/hardware interfaces are specified and reviewed as a matter of course. The PCT Toolkit benefits from these methods by using the interface definitions, along with the system security

model to validate all interactions.

Results of the Toolkit analysis can be organized into tables that can be used efficiently at run-time. Each table, called a privilege control table, contains only the validated interactions and resources for a particular security subject. The operating system [AOS] performs enforcement at run-time. Implementation of the AOS and PCT is discussed further in Section 4.

Invalid interactions are logged for action by a system administrator (or designated integration team personnel during system integration).

3.5 TCB View With System Build

The System Build approach results in a system that minimizes the real-time performance of multilevel security measures. The side effect is that the trusted computing base (TCB) has been distributed and possibly enlarged. Figure 3-2 shows how the TCB is distributed in the System Build approach.

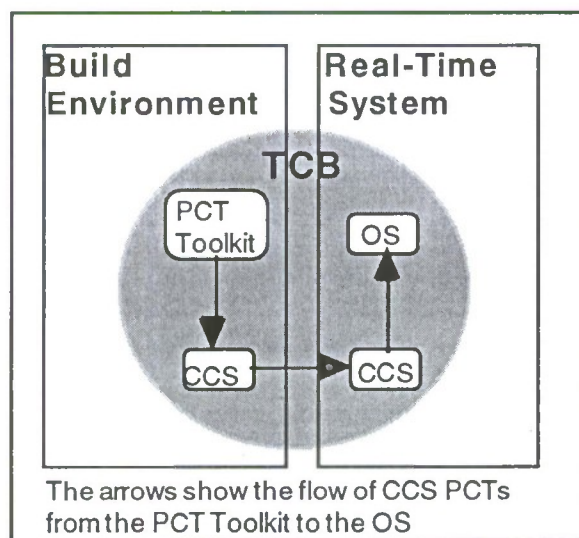


Figure 3-2: The System Build Approach

The separation and distribution of the TCB raises several issues, some of which are raised in [SYSBLL].

One issue is the effect of System Build on certification and accreditation (C&A). C&A of the TCB must be performed on the real-time operating system in concert with the Toolkit. Lack of precedent for this approach impacts the first such system, but should be ameliorated in future instantiations.

Trusted distribution of PCTs from the

build environment, a secure contractor, or government site, to the real-time environment in the embedded system also must be addressed. This problem is solved today by use of a cryptochecksum (CCS), a capability already available in the target system.

3.6 How PCT Toolkit fits into the Software Development Process

Figure 3-3 shows the integration of the PCT Toolkit into the Software Development process. PCT development basically parallels software development. In fact, at each stage the products must be consistent. That is, the source code and the interface information must be consistent or the resulting PCTs and executable software will not be consistent. Inconsistencies can result in a variety of failures. Section 5.3 addresses these consistency issues in detail.

At run-time, when the OS loads a program, it also loads its corresponding PCT. The PCT can only be accessed by the OS. The combination of a software program together with a PCT defines a subject.

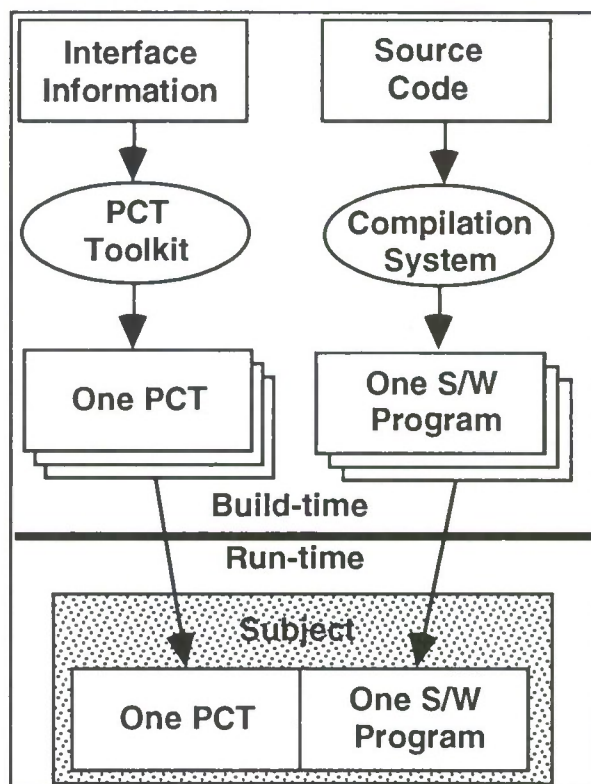


Figure 3-3: The PCT and Code Development Processes Through Run-Time

4. The PCT Toolkit in Detail

The fundamental purpose of the PCT Toolkit is to perform mediation on an information flow model of the system during System Build and to encapsulate the relevant information and privileges into tables that can be used at run-time by the OS, thereby reducing the processing assets dedicated to security in the real-time environment. Figure 4-1 shows the Toolkit data flow.

In order for the PCT Toolkit to accomplish this task it must do the following:

- 1) Understand the Target Environment
- 2) Build an Information Flow Model from Interface Definitions
- 3) Apply the Target Environment Information to Information Flow Model (a.k.a. Consistency Checks)
- 4) Perform mediation using the Bell and LaPadula [BLP] model on events requiring mediation
- 5) Output PCTs and other tables
- 6) Output reports to assist users
- 7) Generate a log file to document all input information, all conclusions (e.g., security violations), and all outputs generated.

In addition to the above, the PCT Toolkit allocates resources within some object classes. As will be shown, this is consistent with its purpose.

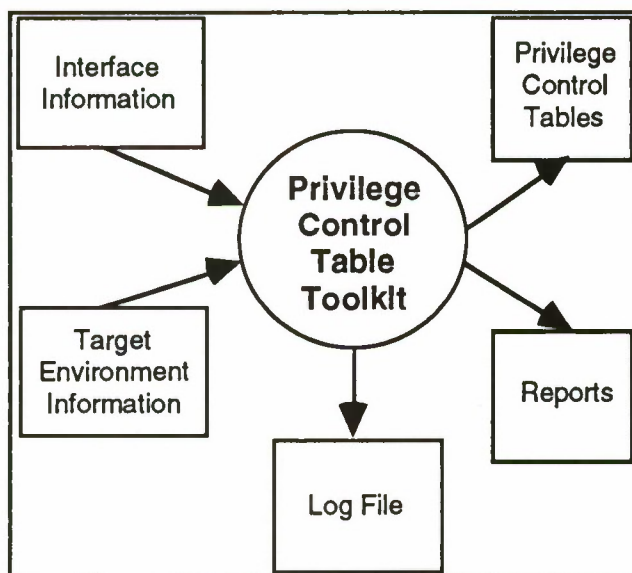


Figure 4-1: PCT Toolkit Data Flow

4.1 Understanding the Target

In order for the PCT Toolkit to accomplish its purpose, it must have knowledge of the target environment. Since the target environment may change considerably from one system to another, the Toolkit must be flexible; therefore, at initialization the PCT Toolkit reads in information that configures the PCT Toolkit for the target environment. This information is contained in configurable knowledge (CK) files. The CK files allow for definition of the hardware implementation, of the security classifications, and of privileged OS services.

4.1.1 Hardware Information

The hardware CK files describe the hardware components and their quantity, interconnect topology, and relevant security capabilities. For example, security characteristics of special purpose processors may be described by the number of allowable programs served, and the ability of the processor to maintain separation of data.

4.1.2 Security Information

One CK file read by the Toolkit defines all classifications in the system and their dominance relationships. The Toolkit has no built-in knowledge about classifications. The security CK files completely defines the semantics of classifications. It supports categories and allows for disjoint classifications to be defined. In addition, as shown in Figure 4-2, it can be defined such that two or more classifications can actually be made equivalent for the purposes of applying the security policy.

4.1.3 OS Information

The PCT Toolkit and the OS are closely tied. As in most OSs, there are some privileged services that are very limited in their use. The subjects that are allowed to use these services are specified in a CK file on a per service basis.

```

-- Classification Set Definitions
--*****
classification_set : LevelSet1
classification      : Unclassified
classification_set : LevelSet2
classification      : Confidential
classification_set : LevelSet3
classification      : Secret
classification_set : LevelSet4
classification      : Secret/SAR/CAT1
classification      : Confidential/SAR/CAT1
classification_set : LevelSet5
classification      : Secret/SAR/CAT2
classification      : Confidential/SAR/CAT2
classification_set : System-High
classification      : System High

-- Dominance Relation assertions
-- (transitivity holds)
--*****
assert : System-High dominates LevelSet4
assert : System-High dominates LevelSet5
assert : LevelSet5 dominates LevelSet3
assert : LevelSet4 dominates LevelSet3
assert : LevelSet3 dominates LevelSet2
assert : LevelSet2 dominates LevelSet1

```

Figure 4-2: Security CK File

4.2 Building an Information Flow Model

After the Toolkit has processed the Target Environment, it can build an information flow model based on interface information defined prior to running the PCT Toolkit. Two methods of gathering interface information have been implemented. The first involves querying a database based on user input. The second involves reading a set of user-specified interface definition files (IDFs).

The database method is driven by a user specifying a set of System Build Capability Tables. There is one table for each capability, each table listing the subjects that must exist for that capability. Based on that list, the Toolkit will query the database for necessary subject and object information.

The IDF method utilizes one IDF per subject, and the IDF contains all object access requests. An IDF is divided into two parts. One defines the subject and the other defines object access requests. In the object section there are subsections for each object type. At least one IDF in the system must identify the object attributes for a given object.

Hereafter, the term IDF shall be used to mean either method. While reading in the information, the PCT Toolkit performs semantic checks, e.g., verifying that every classification used in an IDF is defined in the Security CK file.

4.3 Consistency Checks

The PCT Toolkit checks data consistency to ensure input data are well defined. For example, one consistency check is for readers and writers of volatile objects, that is, objects that exist only when power is present. The PCT Toolkit checks to be sure that there are both readers and writers of every such object. If not, a warning message will be generated. In addition to general data consistency checks, security-relevant consistency checks are also performed by the Toolkit.

The PCT Toolkit will verify that all subject and object names are unique. For subjects, this means verifying that there is only one subject definition. If the Toolkit reads two subject definitions using the same name it will generate an error. For objects, the Toolkit assumes there will be only one definition: if it encounters attributes about an object with the same name, it will check that any attributes specified more than once are identical. Error messages are generated for all inconsistencies.

Some object classes allow only one writer per object. The Toolkit will check that this rule is followed per System Build Capability.

Another consistency check deals with verifying that software is correctly mapped to hardware resources. One or more of the attributes of a subject relate to the hardware needs of a subject. The Toolkit will verify that all such mappings are compatible.

4.4 Supports Resource Allocation

As the PCT Toolkit concept began to evolve from the System Build approach, several potential enhancements became apparent. One such enhancement was global resource allocation, that is, allocation of resources shared by subjects that are not on the same processor. The Toolkit will bind names to specific physical resources.

For example, the Toolkit knows the attributes of all interprogram messages including all senders and receivers, and also has the software/hardware mapping.

Therefore, the Toolkit has all the information necessary to assign bus labels, and its place and role in development process makes it ideal to accomplish this task.

Global resource allocation results in a single PCT object entry that contains the rights of the subject to access the object, and also a mapping of the object to its low-level resource identifier, e.g., a bus label. Because this information is created at build-time and is distributed in a trusted manner via the PCTs, no run-time consumption of resources is necessary to accomplish this task. This enhancement of the System Build approach is consistent with PCT Toolkit's purpose, which is to make security affordable, from a timing perspective, in the system.

The PCT Toolkit supports several types of global resource allocation, although system requirements will dictate which are possible for a specific Toolkit implementation.

4.5 Build-Time Mediation

The Security Policy Model applied by the PCT Toolkit is based on the Bell and LaPadula model [BLP], although another model could be substituted into the tool.

4.5.1 Applying IDF Classification Labels

The interface information contained in the IDFs is labeled according to the actual classification of the information, hereinafter called content labeling. The PCT Toolkit labels information according to its container, hereinafter called container labeling.

Therefore, if an untrusted HIGH subject sends a message, M, labeled in an IDF as LOW, the PCT Toolkit will classify the message M as HIGH. A read request of M in an IDF by a HIGH subject will therefore be allowed. A read request of M in an IDF by a LOW subject will cause a security error; however, the error will be traced back to the offender according to content labeling. Therefore, the error message will state that there was an illegal attempt by a HIGH subject to write a LOW message.

This method is consistent with the model, and yet preserves the user view of the system as described in the IDFs. Furthermore, no information in the IDFs must be artificially overclassified so that the mediation will pass.

The side effect (which is correct) is that information moving in the system outside of

the domain of the PCT Toolkit and its corresponding OS must be protected at the level as classified by the Toolkit.

There are several object classes in the system and the security rules applied to a class may be unique. For example, write access at run-time to some object classes does not return status (write up allowed), whereas for others there is status returned (write up not allowed to minimize covert channels).

4.5.2 Hardware-Specific Checks

Different processors and hardware engines, e.g., a graphics display engine, have different capabilities with respect to security. The PCT Toolkit will use the definitions in the CK file to determine what additional checks must be applied. The Toolkit is not coded for specific checks; rather it determines the correct way to model the hardware and applies the applicable general rule. For example, if the hardware cannot keep different users separate and if it has both read and write capabilities, then the PCT will model the hardware as an object that inherits the classification of its user(s) and to which each user has read/write access.

4.6 Build Tables

Once the PCT Toolkit has built a system model and applied the security policy, it can create the tables necessary for the run-time TCB components to establish a secure state and to enforce the mediation decisions made by the Toolkit.

These tables include not only PCTs but also other tables. This is another case where the PCT Toolkit has taken on greater capabilities than those defined in the original System Build approach.

4.6.1 System Start-Up Information

One output is the System Start-Up Table (SST). At the start-up of the target system, the TCB software that controls the master nonvolatile memory comes up, reads and uses the SST, and then distributes the SST in a secure and reliable manner to each of the distributed start-up programs in the TCB. These programs use the SST to control local start-up. The SST indicates what TCB components should be loaded and the physical resources that should be used during the start-up process. Since the start-up scenario can change based on the System

Build Configuration, and since the PCT Toolkit allocates resources and knows the size of objects, the Toolkit can create the SST that contains information on what is to be loaded, how big it is, and the resources that should be used.

4.6.2 System Manager Tables

The PCT Toolkit also builds a set of tables that are used by System Manager, a high-level program that controls the system, including what System Build Capability is used. System Manager is part of the TCB.

Because the Toolkit knows all programs needed to accomplish a particular capability in the System Build Configuration and knows what hardware is needed, it can build tables to assist the System Manager.

Immediately after start-up, the System Manager program will assess the availability of the hardware assets and based on that will determine which capabilities can be met from the set defined in System Build Configuration. It then chooses the best capability based on predetermined input and/or pilot input. Then, using tables built by the Toolkit, the System Manager can determine what programs and what PCTs need to be loaded. It is the System Manager that communicates with the OS to coordinate loading of a program and its PCT.

4.6.3 PCTs

There is one PCT created for each subject in the target system.

The internal organization of a PCT is very straightforward: there is a header plus one section per object class. The header contains a timestamp, the System Build Configuration identifier, subject information, and other information about the creation of the PCT.

Each object class section is ordered lexicographically by object name, only for those objects accessed by that subject; hence only non-null entries are present in the PCT. The PCT does not contain the object names. In order to save space and to effect an efficient run-time look-up into the PCT, the user program when calling the OS will specify an object by using its relative lexical order for a given object class. Hence, if the object name is first when ordered with the other objects of that class which are accessed

by that subject, then "1" would be specified to the OS.

Unique subject and object identification is contained within each PCT. Each PCT header contains the unique subject identifier. Each object can be uniquely identified by its object class and resource identifier. The resource identifier is either the physical identifier assigned by the PCT Toolkit or it is a system-unique identifier for the class used to allocate run-time resources.

Many object classes allow for frequency of access to be specified, and this information is contained in their object entries. For these classes the OS can perform denial of service. There may also be priority information in each entry for some object classes.

4.7 The Log File and Reports

The PCT Toolkit creates a log file as it executes, which documents all Toolkit operations relevant to a reviewer. This file is always generated and includes:

- A timestamp and other information relevant to configuration management
- A list of all user inputs and options
- All input files read
- All output files generated
- All security violations
- All other errors and warnings

The Toolkit also creates several reports that allow the user to see the information from various perspectives. One report details the information contained in each PCT. There are also reports that organize information by System Build Capabilities. Other reports detail information on all objects or specific object classes. There is also an ASCII table file that organizes all information related to the system information flow model created by the Toolkit. In this way others can build their own tool to read and organize the information that best suits their specific needs.

The Object Cross Reference report is of particular significance. There is one report per System Build Capability. It contains all objects for that capability, together with their object identifiers and classifications. The classification from the IDF information (content labeling) is listed with the classification at which the information must be protected (container labeling).

5. PCT Toolkit Security Issues

5.1 Review Issues

The PCT Toolkit works under the assumption that the information it uses is complete and correct and that it has been reviewed and approved by appropriate personnel. Therefore, if interface information states that a subject needs a specific type of access to an object, the Toolkit will assume that the access request is consistent with the principle of least privilege.

It should be noted that some CK files are more security relevant than others; therefore, the PCT Toolkit uses several CK files so that the review of the Toolkit for certification and accreditation (C&A) is simplified.

CK files were chosen over IDFs for specifying privileged OS services so that this very sensitive information would be located in one small file for easy review. Although there are generally only a handful of such services, they generally give very powerful privileges. Therefore, ease of review was a primary concern. These services are also very OS dependent. This use of CK files keeps the more general IDF information from becoming OS specific.

5.2 Comparison of Input Methods

The database and IDF input methods have different advantages and disadvantages. The advantage of the database method is that the information is logically grouped and there is no repetition that can lead to inconsistencies. Its main disadvantage is that the information used by the Toolkit is not readily reviewable: the reviewer must correctly use database tools to review it, and the review procedures are not well defined. Compounding the review problem is the fact that this method is likely to contain a lot of information that is not of immediate interest to the reviewer. Another disadvantage of this method is that the Toolkit is closely coupled to the database.

The use of IDFs is seen as the better implementation from a security review standpoint: IDFs are easy to review and correspond closely to what is contained in a PCT. However, IDFs require the PCT Toolkit to perform more extensive data consistency checks than the database approach. For example, all IDF references to object O will

be checked to see that all IDFs that define O attributes define them to be the same. One such attribute is the security classification of the object.

5.3 PCT Entry Ordering

The lexicographical ordering of PCT entries within an object class may at first seem rather burdensome since the user must know this order when calling the OS, but it is in fact quite simple to implement using high-level programming languages. For example, in the Ada programming language, the user only need create an enumerated type for each object class. The only burden upon the user is to alphabetize the symbolic names in the enumerated type. One user generates these enumerated type definitions in Ada packages using a tool which accesses the database.

This ordering method has several benefits. It is efficient from both a space and size perspective. It also allows the user to use the same symbolic names as defined in the database. Finally, a side benefit is that if good programming practices are used, the code becomes rather easy to review in terms of the objects to which a program makes reference.

This ordering implementation does not introduce any security problems. The IDF information defines what objects a subject has access to, assuming the mediation checks pass. Hence, if the IDF defines access to five objects of a given class, then the PCT section will contain five entries, with the appropriate access rights. Since all untrusted subjects operate at a single level, any erroneous access within the allowed range will be just that - a programming error. The PCT only contains approved accesses, hence the subject cannot access anything it does not have the rights to access. However, if the subject is trusted and has access to objects of differing classifications, then a potential problem exists; however, trusted programs should be closely reviewed for correctness and any reviewer instructions would explicitly state this as an item for careful review.

6. Conclusion

6.1 Advantages and Disadvantages Summarized

Implementation of the PCT Toolkit has revealed the following advantages of the System Build approach:

- The approach allows us to meet exacting system performance and security requirements simultaneously.
- The approach fits in well with the system and software engineering process, ensuring integrated system security engineering.
- System integration time is reduced due to detection of design and coding errors at system build time. This also reduces life cycle cost by decreasing the number of hours needed for integration in the target environment.
- Run-time reliability is improved as a result of reliability requirements in support of security, and as a result of finding security errors prior to run-time.
- System start-up and initialization is simplified, allowing for faster turnaround and takeoff.

The following drawbacks to the approach have also been identified:

- The approach increases the Trusted Computing Base in size and span.
- It takes longer to perform system updates. Since updates occur infrequently in the current environment, this does not have a significant impact today, but would have to be addressed if the situation changes.
- Changes to subjects/objects and the addition of new subjects requires a system-wide update of the System Build.
- Output of the PCT Toolkit is dependent on the quality and reliability of input data. This is handled today through tool support and development methods (e.g., code reviews).

These drawbacks have been addressed for the current implementation, but to achieve generalization of the approach it is desirable to mitigate these with standard methods.

6.2 Work To Be Done

We believe the viability and benefits of the System Build approach have been demonstrated in its implementation in the form of the PCT Toolkit. There is additional

work that should be done to improve on the current methods, mitigate some of the drawbacks of the current instantiation, and generalize the approach to make it more widely applicable.

The PCT Toolkit was designed to allow for the application of various security policies and models. We believe it would be worthwhile to test this by applying new models (e.g., Rushby Non-Interference) and studying their impact on the tool.

We anticipate that there will be times when a new subject or object will be added to a system. An analysis should be done to determine the viability of adapting System Build and PCT Toolkit for this scenario.

The System Build approach integrates system security engineering with software and system engineering. By extending the integration of the methods and tools, we believe several results can be achieved: increased reliability of input data, improved portability to other software engineering environments, and decreased time to generate a System Build Configuration.

7. References

- [AOS] M.M. Bernstein and C.S. Kim, "AOS: An Avionics Operating System for Multilevel Secure Real-Time Environments," Computer Systems Application Conference, December 1994.
- [BLP] D. E. Bell and L.J. LaPadula, "Secure Computer Systems: Unified Exposition and Multics Interpretation," MTR-2997, The MITRE Corp., Bedford, MA, March 1976.
- [SYSBUILD] J.P. Alstad, T.C. Vickers Benzel, et al., "The Role of System Build in Trusted Embedded Systems," Proceedings of the 13th National Computer Security Conference, Volume 1, October 1990.
- [SYSBLDLL] T.C. Vickers Benzel, M.M. Bernstein, et al., "Real-Time Trust With 'System Build': Lessons Learned," IEEE, 1993.

USE OF THE ZACHMAN ARCHITECTURE FOR SECURITY ENGINEERING

Ronda R. Henning
Harris Corporation
Information Systems Division
Mail Stop W2/7756
P.O. Box 98000
Melbourne, FL 32904
407-984-6009

1.0 Introduction

A system security policy is often perceived as a set of mandatory requirements levied upon the system by an organizational directive or Information System Security Officer (ISSO). To the user, these security requirements may bear little resemblance to his actual working system security policy, which controls data modification and user privileges. In the course of reengineering business processes and information systems, the system modeling activities provide a unique opportunity: This paper presents a methodology for security policy definition using the Zachman information systems architecture as a tool. The system security policy can be extracted from the Zachman framework, providing a technique for reconciling the security policy as defined by directive with the user's working system security requirements.

2.0 Security Policy Derivation -- Today

In the current generation of system specifications, the security policy requirements are often summarized as the requirements for operation in a given secure mode of operation as specified in an organization's security guidance. From the accreditor's or designer's perspective, a system is defined as running in a given mode of operation, and at a given classification level (or range of levels). A system security policy may be divided into the individual sub-policies, for example: identification and authentication, auditing, access control, and network access. Additional specification detail may be presented, such as: the only auditable events are "login" and "logout;" or the system must protect itself from malicious code or virus infestation. Figure 1 illustrates a possible decomposition of a system security policy into subpolicies.

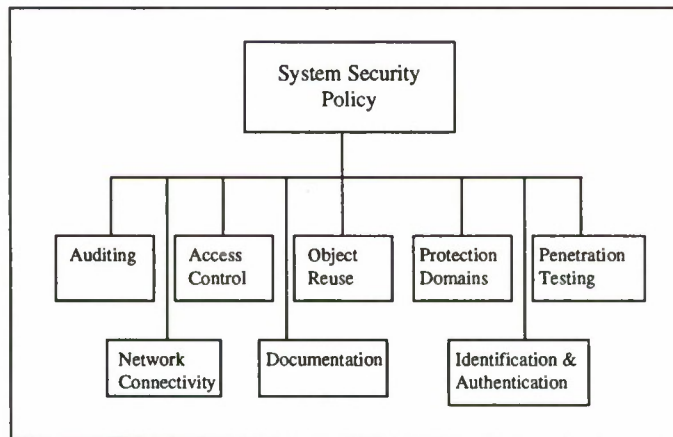


Figure 1. A decomposition of a system security policy.

These requirements may be levied by regulation, not necessarily operational system usage. For example, a real-time command and control system may not require individual user login and logout, citing operational necessity. A mission simulator that provides scenarios for multiple mission planners may not require the flight crews to login/logout. The words "that's not how my system works" usually follow the

realization that organizational policy requirements must be implemented in the re-engineered system. The user does not identify with the requirements, and perceives security as an obstacle to his mission..

2.1 The Hidden Security Policy

Beyond the explicit security section of a system specification, there may be an implicit security policy. This policy is often couched in phrases such as

- “ensure operators can only perform operations X, Y, and Z,”
- “only mission managers can modify plans,”
- “there are only two roles in the system: operator and user.” and
- “users cannot change system parameters”.

This implicit security policy must be elicited by gathering these phrases from the specification and concept of operations documents. Unfortunately, the end user of the system does not always recognize that the implicit security requirements are security requirements. When an architecture uses Commercial-Off-The-Shelf (COTS) security mechanisms to implement implicit security requirements, the customer may intervene. For example, an access control requirement, implemented through the use of operating system access control lists, a standard COTS mechanism, can be met with a customer comment such as: “That can’t be a security requirement. Security didn’t specify it. The old system has some software written to check if the user is allowed to modify that file.” This approach results in sub-optimal solutions with no inherent system flexibility. In such a scenario, the working security requirements can only be solicited if the data flows are defined and analyzed, placed in the context of the updated system requirements, and then designed into the new system. This approach provides the functionality required by the user, with security mechanisms that can be maintained throughout the system lifecycle. System certification time is also decreased, because it is much easier for a Designated Approval Authority (DAA) to inspect the configuration of access controls within a system as opposed to code inspection of new security mechanisms.

2.2 Security Requirements Synthesis

When the implicit security requirements are coupled with the directive-based security requirements, a true baseline of security requirements for a system emerges. At this point the secure systems engineer applies risk management techniques to determine the relative criticality of the data. This information helps define what protection mechanisms to apply in any given system architecture. For example, if a system processes five percent of the data at the classification TOP SECRET, and the remainder of the data is UNCLASSIFIED; it may be much more cost effective to build a subset of the system to address the TOP SECRET data segregation requirement. In conjunction with the customer, the security architect for any system must derive:

- which data elements to protect,
- how much protection this data requires,
- how this data may be modified, and
- how this data is communicated to other systems.

The captured information is discussed in the Security Accreditation Working Group, and documented in the Security Certification and Accreditation Report. It is used by the DAA to define the security characteristics of the system, and to determine if appropriate safeguards were applied. From the DAA’s perspective, the system architecture must provide protection consistent with the data contained in the system.

Presentation of the information in a cohesive format is the responsibility of the security engineer. Security organizations speak in terms of subjects and objects, with Mandatory Access Control (MAC), Discretionary Access Control (DAC), Identification and Authentication (I&A), and Object Reuse policies, using terminology that is unfamiliar to the majority of system customers. This can make it difficult to reconcile requirements as expressed by the user in terms that are understandable to both the customer and the certification organization. To facilitate this task, the Zachman Model of Information Systems Architecture can be used. The Zachman Model is normally applied to general purpose information modeling tasks. With some forethought, it can readily be adapted to incorporate security policy modeling as a part of traditional information modeling activities.

3.0 The Zachman Framework

The Zachman Framework for Information Systems Architecture (ISA),¹ defined in 1987, is a logical construct to define and control the interfaces and integration of all components of a system. The framework of the Zachman model enables systematic capture of system specific information from the various perspectives with respect to a system architecture. Figure 2 illustrates the 30-cell Zachman model, tailored to support an information systems re-engineering application. In this customization of the model, the system developers have an existing operational system in place. The model is applied to capture the security policy of the existing system to ensure the actual user requirements are understood prior to system re-development. When this framework is complete, the explicit, directive based security requirements can be applied and overlayed into the framework, reconciling the implicit, working model and the directive based model for the system's security requirements.

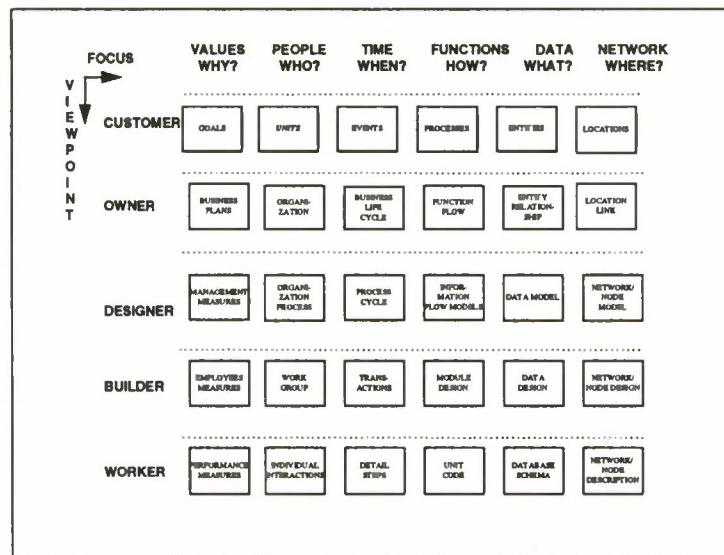


Figure 2. The Zachman Framework for Information System Architecture.

The Zachman framework has two very distinctive features that make it ideal for information modeling. The framework may be applied at any level of abstraction in the system development process, from a global enterprise, to a system, subsystem, or major module level. The framework also gives the modeler latitude in that any data representation technique can be used to model the inner workings of each cell. For example, entity relationship diagrams, IDEF (Integrated DEFinition language, or ICAM (Integrated Computer-Aided Manufacturing) DEFinition Language)² models, and conceptual graphs are all equally valid representations of the information contained within a given cell.

As one changes perspective from the customer level down to the worker level, more detail is provided, and less large scale perspective from the upper cells is visible. The system model becomes more implementation specific. However, the requirements traceability between layers can be maintained through backward references to upper layers of cells. This traceability is critical in security requirements engineering, where tracing a global access control requirement may translate into explicit setting of access controls on specific files or directories within an operating system.

¹ The Zachman Model was initially described in "A Framework for Information Systems Architecture," IBM Systems Journal, Vol. 26, No. 3, 1987, pp. 276-292.

² The IDEF model is described in various publications, including: "IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications." Richard J. Mayver Ph.D., Michael K. Painter, and Paula S. deWitte, Ph.D., Knowledge Based Systems Inc., 1993, and "The IDEF Framework Version 1.2," publication of IDEF Users Group Working Group 1 (Frameworks), May 22, 1990.

The framework provides a taxonomy "that helps us understand the perspectives of various players in the development of an information system and the descriptions of the system that can be produced during its creation."³ The model is frequently used as a framework during information systems re-engineering activities to support the solicitation, identification and mapping of the following information associated with an information system's:

- goals, objectives and environment,
- customers served,
- time constraints,
- functional description,
- information architecture, and
- supporting infrastructure.

Application of any model implies a set of rigor and structure. For the Zachman Model of Information Architecture, the basic structural framework rules are:

- The columns in the framework have no order, which would create a bias towards one perspective of the system over other perspectives.
- Each column is based on a simple, basic modeling technique. The columns provide answers to the basic "who, what, when, where, why, and how" questions.
- Columns are unique, that is, their contents are not repeated, which preserves the ability to define a categorization scheme for the model.
- Rows represent a distinct, unique perspective of one of the models (i.e., scope, enterprise, system, technology, component, or working system).
- Each cell is, in itself, unique. So the resulting metamodel is, in itself unique.
- The composite, or integration of all cell models in a single row constitutes a complete model from the perspective of that row.
- The logic is recursive, allowing increasingly more detailed models to be developed.⁴

The resulting information system architecture provides a unique model, where, at any given row level, an integrated perspective of the system can be produced answering "who, what, when, where, why, and how." The framework allows ownership of activities and data to be established, and traced throughout the system development process.

In short, the Zachman Information Systems Architecture can provide a consolidated view of a system, to whatever level of detail a modeler chooses.

3.1 Application of the Zachman Model

Within Harris Corporation's Information Systems Division, an Information Systems Reengineering Action Team was tasked with the definition of a corporate information systems reengineering methodology. The methodology created is based on the Zachman Model, and is used to define the present system, the desired system, and a transition strategy to bridge the user's expectations between the two system models. In the absence of a commercial off the shelf solution, Harris developed a middleware application that automates the support of the Information Systems Reengineering Methodology. The middleware application provides built-in solicitation for the development of Zachman cell contents. It also supports requirements management by enabling the mapping of the requirements to the same frame of reference, resulting in a requirements repository reflecting the current system and its evolving replacement. The tool does not replace sound engineering discipline, but facilitates requirements capture and interface definition activities. Figure 3 illustrates the top level menu of the Information Systems Reengineering Task Management Tool.

³ Bruce, Thomas A., "Simplicity and Complexity in the Zachman Framework," *Data Base Newsletter*, May/June 1992, p. 3.

⁴ Sowa, J.F. and Zachman, J.A., "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No. 3, 1992.

As Is Project Records

CELL NAME: **FUNCTION FLOW**

Cell Reference: **D2**

Focus: **Function (how?)**

Viewpoint: **Owner**

Cell Owner: **ISRAT**

Date Of Last Update: **5/5/95**

Inputs: **Description of customer prepared and enterprise plans (C2)**

Outputs: **Functional models (D3, E2)
Enterprise data dictionary (D3, E2)**

Cell Definition: **At this point, the functional flow model of the business needs to be identified. A model of each of the business processes needs to be established in descriptive terms such as input - business process -**

Questions: **What are the business processes associated with the enterprise?
What are the inputs and outputs associated with these business processes?**

Success Criteria: **Diagram the enterprise context.
Identify and label each input, output, and external interface of the enterprise.**

Supporting Tools: **Word PwrPnt Access Excel N Cancel Tools**

Navigation: **TOKEN PARKING**

Print Report

Return

User Feedback

Figure 3. Information System Requirements Modeling Tool Screen.

In the course of using the tool for generic requirements management, it became evident that applying the Zachman Model to security engineering would be a relatively uncomplicated application. With some minor rework, the model could be readily adapted as a security policy modeling tool, providing a framework for the reconciliation of the implicit and explicit security requirements associated with a system architecture. It could also provide a useful tool to the system security certification team as a requirements traceability matrix. Applied in this manner, the Model traces the top level system requirements specification down to the actual implementation mechanism.

4.0 Security Modeling Integration with the Zachman Model

The Zachman Model cell organization is structured into five levels, or rows, representing increasingly detailed perspectives on the system in question, as defined in the following table.

Table 1. Zachman Model Cell Organization from a Layered Perspective.

Layer	Perspective	Description
1	Customer	Defines a clear and coordinated boundary (domain) of the system for the purposes of identifying people, subsystems, and needs impacted by the system.
2	Owner	Captures the business and organizational relationships, and their external interfaces. Documents requirement sources, including those derived from legacy systems.
3	Designer	Defines functional capabilities of the system and establishes and documents the architectural foundation for system design and development.
4	Builder	Establishes and documents the architectural design. Provides basis for system measurement.
5	Worker	Provides detailed description of design and methodology for monitoring and correcting system performance.

For security policy modeling purposes, the first three levels of the perspective hierarchy (customer, owner, and designer) are extremely useful. They provide the consumer perspective of the system's end

user, the perspective of the system "owner" or contracting entity, and the perspective of the designer, or systems engineer. In other words, the "as built" and used in daily operation perspective, the "as desired" operation perspective, and "as actually specified" perspective.

One of the more common modeling methods that can be used to define cell content is the IDEF language. The IDEF model, layer 0 (IDEF0) model provides a representation of the inputs, outputs, controls, and mechanisms associated with a given cell. An IDEF0 model of the inputs, outputs, and process constraints associated with each cell can generate additional security relevant information. Figure 4 illustrates the generic IDEF0 model using an external perspective to the cell itself.

Without any additional information, the external perspective provides the data flow through the system, the command media flow down from upper levels, and the mechanisms with associated system performance constraints.

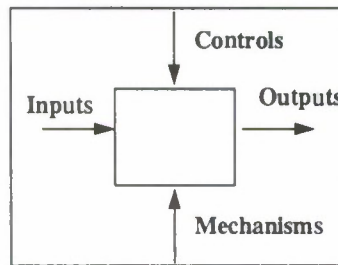


Figure 4. Generic IDEF0 Model- External Perspective.

4.1 Security Derivation

When the top three layers of the Zachman model are applied to a system, without any additional security information, the security engineer can readily obtain:

- the system functions,
- the system information flows,
- the network connectivity of a distributed system,
- the data model,
- the data "owners, modifiers, and users," and
- the responsibilities of organizational entities associated with the system.

If some additional security relevant information is appended to the IDEF0 model constructs, the external perspective illustrated in Figure 5 results, and the model construct becomes more useful for security policy modeling. Annotation of the IDEF model with this minimal additional information provides the security engineer a more robust picture of the potential security problems associated with a system. For example, the customer can claim the system does not in any way, shape or form connect to a system at a higher classification level. If a particular input can be identified as coming from a particular source system, the classification level associated with the source system can be verified. Similarly, a list of user roles or access control rules such as "only users at location X" can be assimilated into the system access control policy. Determination of possible information downgrade procedures can be determined by examining inputs, outputs, and mechanisms for classification.

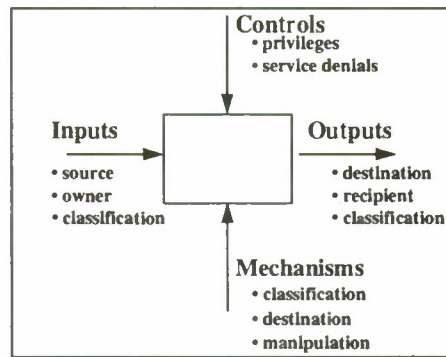


Figure 5. Annotated IDEF0 Model with Security Attributes.

The objective of this exercise is to find possible inconsistencies between the system as described, as specified, and as mandated in governing policy documents as early as possible in the system specification and design process. The goal is to have an accurate representation of the system consistent with applicable security policy and doctrine in effect.

4.2 Applying the lower model layers

While security policy modeling itself is not as concerned with the builder and worker levels of the model, these levels have great value as a security accreditation aid. Through the designer perspective level, a complete set of system security requirements and a security policy specification is defined. Continued application of the model at the builder and worker layers provides requirements traceability down to the level of security mechanism implementation. Table 2 illustrates this correspondence.

Table 2. Correspondence of Upper Layers to Lower Layers.

Level	Perspective	Description
1	Customer	Tasking is received in the system from "System Y." Crew Chiefs read the tasking and come up with how to fulfill it.
2	Owner	Mission plan is developed by "Crew Chiefs" only.
3	Designer	Mission plans are kept in individual text files. Only "privileged users" (i.e., Crew Chiefs) can modify the mission plans.
4	Builder	Use system discretionary access control to define modification privileges to the file.
5	Worker	Access Control List entry shows "write access"; audit trail record written for all file access attempts.

By providing visibility and traceability across the security requirements, it may be possible to more readily develop system security test plans and ensure comprehensive coverage of the requirements.

4.3 Near term Additlons

Security requirements can be levied upon system architectures from directives and policy guidance documents such as Director Central Intelligence Directive 1/16 (DCID 1/16) and the individual service's security directives. These guidance documents superimpose a set of static requirements upon a system, defining the requirements for a given mode of operation. For example, a system high mode of operations requires individual user identification and authentication. It would be most useful if such static requirements could be incorporated into the basic model template of thirty cells and loaded with the initial Zachman model definition. This would avoid duplication of effort, and eliminate the possibility of "overlooking" a requirement. One template for each mode of operation and each policy directive would be required. Current activity addresses the decomposition of the mode of operation requirements into appropriate cells

in the architecture. Then a menu selection of mode of operation will be possible in conjunction with initial population of the hierarchy.

The IDEF0 template fields prompting for input, output, control, and mechanisms could be modified to address the security annotations discussed above. Again, this activity is designed to integrate the security process with the fundamental information and process modeling activities associated with the system. The goal is to make security an integral part of the system design and development activities as painlessly as possible, with minimal impact on the customer and the engineering staff.

Use of the model output as accreditation evidence has not been attempted. Its acceptance as accreditation evidence would be contingent on the diligence of model maintenance. The initial modeling could be translated into a system security policy document in the traditional sense. Doing the model during requirements specification, and not maintaining correspondence between the various model perspectives during system development would make it difficult to submit the model itself as credible as accreditation evidence. If the model is maintained, and traceability among the cells can be demonstrated to the satisfaction of the Designated Approval Authority, the model should be acceptable as part of a system accreditation package.

5.0 Problems in Use

Users of the Zachman modeling methodology have previously discussed the importance of maintaining a consistent perspective on the system across the model⁵. It is easy to become so intent on modeling a given cell that great detail is applied, and other cells may be ignored or minimally addressed. In this case, one of two things has happened: either the customer has provided great detail in his description of one part of the system, or has provided minimal data about the minimally addressed cells.

The recursive nature of the model makes it possible to define complete iterations of the model at varying levels of complexity. In this scenario, one could start with a top layer model of the Air Force, with subsequent layers for major command and control systems, their subsystems, the subsystem's subsystems, etc. As with any information modeling technique, the practitioner must know when the costs associated with modeling outweigh the benefits.

Another problem with the model, particularly when used in association with an automated tool, is that its use is often considered a "short cut" to requirements engineering. The model does not replace requirements engineering in complex information systems. Rather, it is a disciplined approach to manage the complexity of a system and its requirements. Applied in the context of security engineering, it affords a technique to graphically illustrate and manage the security requirements associated with a system architecture.

6.0 Conclusions

In conclusion, the Zachman Information Systems Architecture framework for systems modeling provides a commonly used technique that can be applied to security policy modeling early in the system requirements definition process. By applying the top three levels of the Zachman hierarchy, it is possible to develop a descriptive security policy in simple English that can be understood by the system consumer organizations. Annotations to the IDEF0 model for classification, source, destination, and data manipulation constraints allow rapid location of possible problem areas before they are designed or implemented in the system architecture. Use of the lower layers of the model provides additional traceability that is highly useful to the Designated Approval Authority as part of the system security certification evidence. As such, it is a valid tool to apply to security policy modeling when developing an information system. Application of the Zachman Model provides a technique to:

- express doctrine oriented security requirements,
- reconcile these requirements with the "as built" security requirements, and
- provide traceability for requirements from specification to implementation.

Ideally, incorporation of security requirements into the framework should result in a more integrated approach to security requirements analysis, with the eventual inclusion of security requirements engineering into conventional systems engineering as an integrated requirements engineering activity.

⁵ Sowa, J.F., and Zachman, J.A., "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No. 3, 1992.

7.0 Acknowledgments

The author is grateful to Adele Park, Tony Whalen, Eric Meijer, Mary Englert, and the Harris Information Systems Reengineering Action Team for providing their resources, thoughts and comments to the preparation of this paper.

8.0 References

Bruce, Thomas A., "Simplicity and Complexity in the Zachman Framework," *Database Advisor*, pp. 3-11, May/June 1992.

Holbein, R., Teufel, S., and Bauknecht, K., "A Formal Security Design Approach for Information Exchange in Organizations," *Proceedings of IFIP Working Group 11.3 Ninth Annual Working Conference on Database Security*, Rensselaerville, NY, pp. 291-317, August 13-16, 1995.

IDEF Users Group, *The IDEF Framework*, Version 1.2, IDEF-UG-0001, May 22, 1992.

Mayer, Richard J., Painter, Michael K., deWitte, Paula S., "IDEF Family of Methods for Concurrent Engineering and Business Re-engineering Applications," Knowledge Based Systems Inc., 1993.

Pickett, R., "Process Modeling through IDEF, a White Paper on Applied Information Technology," 3 December 1993.

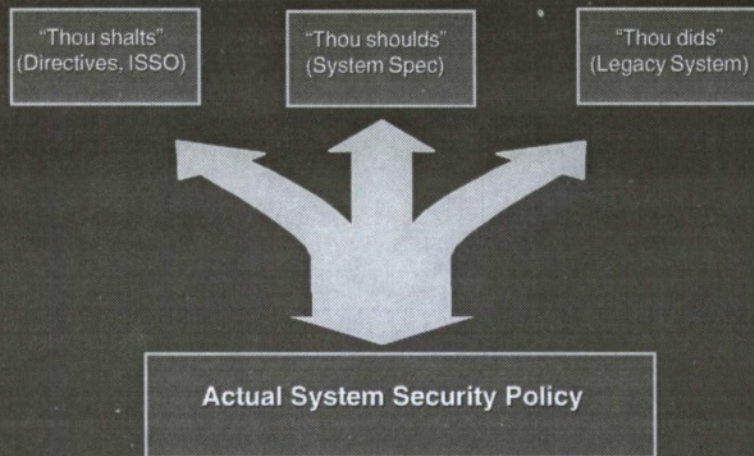
Sadowski, A., et al., *Enterprise Management Analysis*, Enterprise IMS TR4191-01, July 15, 1993.

Schoch, D.J., and Laplante, P.A., "A Real-time Systems Context for the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 34, No. 1, pp. 20-38, 1995.

Sowa, J.F., and Zachman, J.A., "Extending and Formalizing the Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 31, No.3, pp. 590-616, 1992.

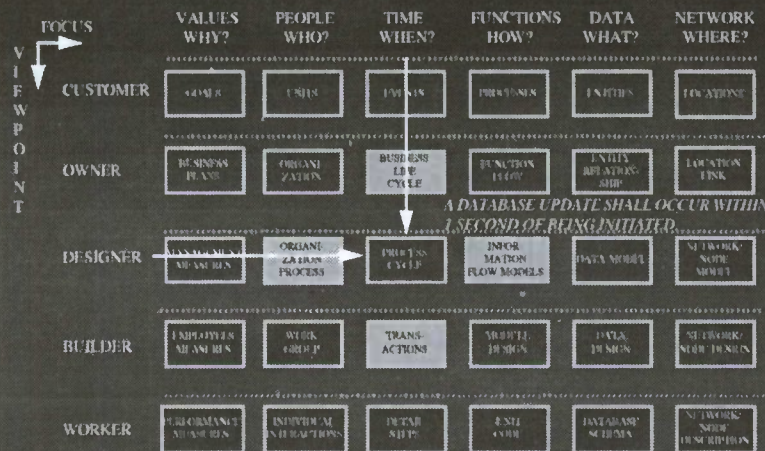
Zachman, J.A., "A Framework for Information Systems Architecture," *IBM Systems Journal*, Vol. 26, No. 3, pp. 276-292, 1987.

Security Policy Derivation



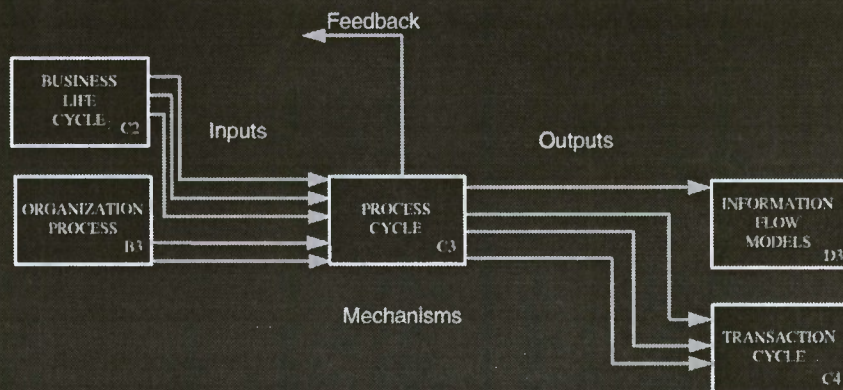
Ronda Henning - 1

The Zachman Framework



Ronda Henning - 2

IDEF0 Model Cell Example



Ronda Henning - 3

Security Relevant Information

Model Construct

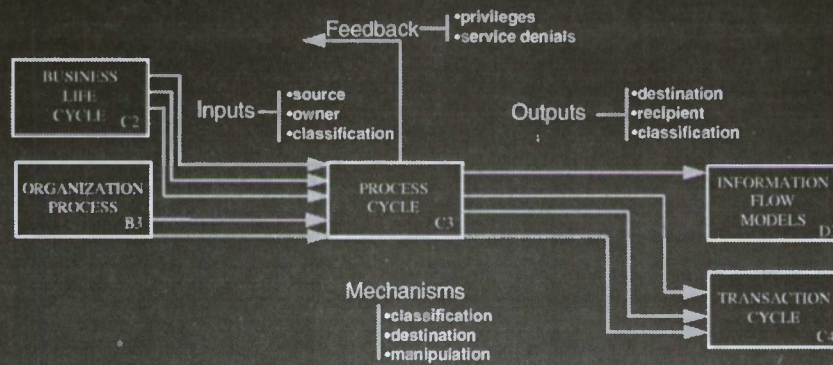
- System Functions
- Information Flows
- Network Connectivity
- Data Model
- "Owners, Modifiers, Users"
- Organization Responsibilities

Security Translation

- What it does
- How Data Moves in the System
- What other systems it talks to
- How Data is Organized
- Who does what to what data
- Who "owns" the data and system processes

Ronda Henning - 4

IDEF0 Model Cell -- Security Annotated



Ronda Henning - 5

Security Information -- Annotated

Security Issues

- Classification problems
- User Roles
- Access Control Rules
- Downgrade Policies

Examples

- Mismatched system classification
- Operator, Administrator, User
- User with "X" privilege can do "Y"
- Message release with human in loop only

Ronda Henning - 6

Developing Secure Objects

Deborah Frincke
Department of Computer Science
University of Idaho

Keywords: Object-oriented design, development, security policy, COTS

Abstract

Distributed object systems are increasingly popular, and considerable effort is being expended to develop standards for interaction between objects. Some high-level requirements for secure distributed object interaction have been identified. However, there are no guidelines for developing the secure objects themselves. Some aspects of object-oriented design do not translate directly to traditional methods of developing secure systems. In this paper, we identify features of object oriented design that affect secure system development. In addition, we explore ways to derive secure object libraries from existing commercial off-the-shelf (COTS) class libraries that lack security, and provide techniques for developing secure COTS libraries with easily modifiable security policies.

Introduction

Object-oriented design (OOD) techniques have become one of the more popular methods used in designing software systems. Some reasons for this popularity are perceived benefits such as a high potential for software reuse, improved reliability, and lower development and maintenance costs. In particular, distributed system design seems especially amenable to an object-oriented approach. Object-oriented programming (OOP) languages also appear to be very appropriate candidates for use in secure system development. OOP languages usually support information hiding, which assists in designing software components that separate policy and mechanism and more reliable software.

There are, however, pitfalls. Object-oriented systems are designed using individual objects that are actively responsible for maintaining their own integrity. It might not be appropriate to design such systems using the traditional 'security monitor' paradigm. Newer OOD techniques, such as Gamma's design patterns, use dynamic modification of system components, where functionality can change substantially as the system runs. If components can change functionality dynamically, it may become quite difficult to validate or ensure maintenance of security policies. Object-oriented systems are intended to be assembled using pre-existing components, and these components may not have been developed for use in secure environments. Potentially, every object in a distributed object system is vulnerable to attack or misuse. Thus, secure system designers must pay particular attention to the security-relevant attributes of the objects they use. Secure system designers who wish to take advantage of OOD will need to address these issues.

In this paper, we identify some features of OOD that affect secure system development. We also explore ways to derive secure object libraries from existing commercial off-the-shelf (COTS) class libraries that lack security, and techniques for developing secure COTS libraries.

Background

Many features of OOP languages, such as their support for information hiding, data encapsulation, and separate components, make them seem well suited for use in secure system development. OOD techniques potentially can make it easier to separate software system's security policy from the mechanisms used to enforce it. Systems with a high degree of modularity should be easier to validate than those without it. Undesirable information flow should be easier to prevent by data hiding techniques. In this section, we describe standards and issues for secure systems, discuss what OO developers have done to address these issues to date, then describe some features of OOPs in general and C++ in particular that are relevant to system security.

TCSEC

The Department of Defense' Trusted Computer Security Criteria (TCSEC), was developed to serve as a guideline for secure system developers, as a way to specify assurance requirements for procurement, and to provide an assessment standard for secure systems. TCSEC descriptions refer to an abstract *security monitor*, so access to system resources is (abstractly at least) guarded by a centralized monitor. This abstract description has been used in implementing secure systems, partially because it is easier to validate correct resource access control with this design. However, if individual objects are actively responsible for maintaining their own integrity, the mechanisms responsible for enforcing object access policies are likely to be distributed throughout the system, rather than gathered in a centralized location. This distribution may make the validated assurance requirements more difficult to meet.

CORBA

One emerging standard for distributed systems is the Common Object Request Broker Architecture, or CORBA. The CORBA standard has been developed through the Object Management Group (OMG) Consortium, which includes over 500 members. CORBA's Object Management Architecture (OMA) describes how end-user application objects, object services, and general purpose services interact together within a distributed object system. Cooperation is achieved through the Object Request Broker (ORB), which enables objects to transparently send and receive requests to other objects [MZ95].

At the time of this writing, the CORBA standard itself discusses security only briefly, although this is expected to change in early 1996. CORBA requires authentication of object clients but does not describe how this will take place [MZ95]. The Basic Object Adapter is responsible for defining how objects are activated (shared server, unshared server, server-per-method, and persistent server), and includes the following five functions as described in [OHE95]: *Authentication* of object clients, although the style (and trustworthiness) of this authentication is not defined, *Implementation Repository* for installation and registry of objects and object descriptions, *Mechanisms* for object activation/deactivation, communication with objects (including parameter passing), and generating/interpreting object references, and *Activation/Deactivation* and *Method Invocation* of implementation objects.

In 1994 the OMG issued a *White Paper on Security* to address the issue of security in distributed object systems and provide guidance to OMG members in their development of proposals

for security in CORBA-compliant object systems [Gro94]. This white paper outlines aspects of distributed object systems which can affect the security of the overall system, such as the number of components, complexity of component interactions, and lack of clear-cut boundaries of trust. Specific security areas of interest include confidentiality, integrity, accountability, and availability; functional security requirements include identification and authentication, authorization and access control, security auditing, secure communication, cryptography, and administration. OMG is reviewing several proposals from vendors such as Hewlett-Packard and Sun.

Our work evaluates methods of implementing security issues such as those discussed in the CORBA security standards. We identify general principles for secure object development which are expected to be important both for customized security components that do not adhere to CORBA standards and for components that are developed for use within CORBA.

Low level design effects

Design perspective is of particular concern because it has such significant effects on policy, and these decisions are made by class developers (and thus unlikely to be identical between competing class libraries). Changing libraries or adding objects from new libraries may result in a different access control policy. Secure system designers cannot take advantage of “interchangeable components” without carefully considering the details of the design strategies used in developing the class libraries they use. In this section, we summarize ways in which decisions made by low-level component designers can change the security policy of the system.

Identifying subjects and objects Most access control models require the subject to have a security label that is compared to that of the object. The Bell and LaPadula access control model (BLP), a subject may obtain data (read from) an object only if subject's access control label dominates¹ that of the target object. If the subject wishes to write to a target object, the *object's* label must dominate that of the subject. In traditional operating system activities, it is clear which participant is the subject and which is the object. This determination can usually be made based on the *type* associated with the participant. When a process requests information from a file, the file is the object and the process is the subject. Files will rarely if ever be subjects, though processes may be either subjects or objects. Due to the emphasis on active objects OOD techniques require, traditionally passive objects such as files may become subjects in OOD systems. Designers must determine whether a File object that *updates itself* and *provides data about itself* is really the same as a passive file that is written to or read from.

Placement of enforcement mechanisms Consider a file containing records, each with a security label. Traditionally, access checks take place at the file level, perhaps handled by a file monitor. A more active design places responsibility with the record, as the object most closely concerned. Rather than encapsulating policy enforcement in a single external monitor, in OOD we are more likely to distribute both the responsibility for enforcing policy and the mechanisms for enforcing policy. Thus, low-level object designers will take on responsibility for system security design. For example, consider a file containing records, each with a security label. When clients request records from the file, should the *file* authenticate and validate the request before producing the requested

¹ *Dominates* here means that in the universe of access control labels under discussion there is a partial ordering $O \prec S$ between some object and subject labels, and the rest are unrelated. Usually equality of labels is included.

Activity	Result
Lists copy themselves at client's behest. Unlabelled elements.	Only elements readable by client
Lists copy themselves at their own request.	Identical lists
Unlabelled lists pass copy request from client to labelled elements, which copy themselves	Only elements readable by client
Unlabelled lists tell list elements to insert themselves in the new list	Identical lists
Labelled lists pass on copy request from client to to labelled elements, which insert themselves in the new list	Only elements at the same level as the client

Figure 1: Copied object characteristics

items, or should the *records* decide whether a client may perceive their value? If data is to be written to the file on the behalf of a user, do we consider the data's label or the user's label?

Copy operations OOD places copy operations within class definitions. An OOD list copy is designed so that the list being copied is responsible for performing the copy itself. Table shows how result of a copy operation changes depending on whether the subject is considered to be the list being copied or the client requesting the copy operation. This in turn depends on how 'active' the list's elements are. This decision would ideally be consistent for all components used within a particular system, but that is unlikely to be the case if these components are obtained from different COTS libraries.

Aggregations When low-level objects are responsible for determining the extent of client's access to their contents, designers will need to take care to shield the existence of some objects from clients, lest they introduce a covert channel. A simple example involves a list, a client, and a set of list elements. The list's designer must balance OO design strategies (making objects responsible for managing their own contents) with security needs (such as preventing inappropriate information flow). Suppose that the designer decides to create a secure list. One option is to give the list's elements control over their own values. Suppose that a client object requests information from a list element. Since the element object is active, the element may refuse to supply information, but cannot prevent the client object from learning of its existence (potential covert channel). An alternate design may be used to eliminate this covert channel. Secure lists may be written so that not only do the list elements know their own security labels, but the list is the object that determines whether the client will know if an individual list element exists at all.

An advantage of placing access controls at a low level (say, with data elements) is that elements may be passed between clients without concern. If policy enforcement is at too high a level (say, file level), then the client must be trusted not to pass high security records to low security subjects. If the security policy enforcement is lower, then the client object need not be trusted.²

² Assuming objects really do control their own integrity and information flow. In languages that allow programmers to bypass the high-level visibility constraints, for example using pointers, additional methods are needed. Objects might include digital signatures to prove that they have not been modified in transit, and information might be encrypted.

Existing libraries

If OOD is to be fully exploited by secure system developers, they should not expect to design all of their system objects in-house. However, if externally developed libraries are used, it will be necessary to add security features to the classes they contain. In this section we explore ways to derive secure object libraries from existing commercial off-the-shelf (COTS) class libraries that lack security.

We consider a collection of interacting objects to be a system. Objects within the system may be active or passive or both. Thus, we need to define both an object system security policy and provide a set of object system security mechanisms. General goals for secure object design follow:

1. Attach an identity to each object and object client
2. Separation of policy and mechanism
3. Separation of object abstractions and security requirements
4. Information Flow and Covert Channels

Without supplying a unique and immutable identity for each object in a distributed system, it may prove difficult to reliably determine whether the interactions an object requests should be allowed. Proper identification and authentication is required for most levels of secure systems.

Separation of policy and mechanism is important for several reasons. There is no single “correct” access control policy that can be predetermined. If components are to be reused in different systems, it is important to make policy modifications easy to perform. Second, there is no single “correct” set of mechanisms that can be determined in advance. Some mechanisms are expensive, unavailable, or illegal.³ Finally, changing mechanisms can effectively change policies, and this would be unacceptable in most secure systems.

If object abstractions are too closely connected to system security requirements, maintaining the system will become quite difficult. First, class designers will be required to become cognizant of security concerns. Second, changing security requirements might result in forced changes to all classes in the library, which would add considerable expense as well as increasing the likelihood of error. Finally, validation would become harder, since security requirements would be hard to distinguish from object characteristics.

Direct modification

We can create secure objects adding security features to their definition. Although direct modification should not really be considered a component reuse technique, we include it here for completeness and comparison. Direct modification should ensure at least the following: the object has an appropriate sensitivity label, all public members manipulate data in accordance with that label and the label of the client object. In our example, the functions that manipulate the `File` objects data should be declared `private`, and hence invisible outside `File`. Public functions, `readSecureFile()` and `writeSecureFile`, would be made available for use by `File` clients. This

³Ex: certain forms of cryptography used in authentication mechanisms are illegal to use or export in some countries, including the USA.

method is simple and efficient, and the security policy associated with each object is easy to determine and modify.

Unfortunately, since the security policy is embedded within the class, changing the system security policy implies changing all class descriptions, as does a change in the choice of security mechanisms. Validating this security policy implementation will be time consuming, since we must examine every class. The security mechanism is necessarily visible to clients.⁴ and we are assuming that the calling object will properly supply the client's identity.⁵ In this particular example, `File` data will not retain the original security label once passed to a client, making it difficult to enforce fine-grained information flow controls or to retain access control information across boundaries.

It is often impractical or impossible either to modify class descriptions or to create secure versions. The former may cause existing software to break, and the latter will certainly increase application code size (as well as being tedious!).

Wrappers

We can use wrappers to provide a secure interface to objects that are not already. This technique is expected to be most useful when portions of the class definition or implementation cannot be changed or are invisible, as with proprietary or compiled libraries. In this technique, we use an existing library of objects that does not include security information, and supply a wrapper class that contains security-relevant data such as identifiers and sensitivity labels as well as enforcement mechanisms.

`SecureFile` is defined using private inheritance from the `File` base class, which makes `File` members invisible outside `SecureFile`, and adds operations that manipulate the base class securely. `SecureFile` clients cannot manipulate the file unless they pass the authentication test supplied by the wrapper. The COTS vendor can modify the `File` base class without affecting the security policies or mechanisms of the specialized class. In particular, `File`'s designer need not consider whether `File` objects will be used in a secure environment. Additional member functions could safely be added to `File`, e.g., `char *appendFile(char *data)`. Because private inheritance is used in creating the secure class, such additional member functions will not be available to users of `SecureFile` unless the `SecureFile` designer adds members that use them, and so the read/write policy of `SecureFile` cannot be bypassed.

There are still disadvantages. We cannot authenticate `SecureFile` clients without adding wrappers to those clients as well, and `SecureFile`'s designer must be cognizant of changes in the underlying class `File` to maintain equivalent functionality. This latter point would not be an issue if we had used public or protected inheritance from the base class rather than private, but the first of these would permit clients of `SecureFile` to bypass the security mechanisms, and the second would let descendants of `SecureFile` bypass those mechanisms. This method is somewhat more suitable for communication between trusted objects and untrusted objects, since untrusted objects cannot manipulate trusted data directly without backwards type casting of the secure object.

Multiple base classes

⁴The implementation code is visible unless we pre-compile member function code and only make the class definition header visible. While "security by obscurity" is insufficient for data protection, sometimes it is desirable to hide security mechanisms. That is not possible here unless the entire class definition is hidden.

⁵Acceptable within a trusted object base, unacceptable for untrusted objects.

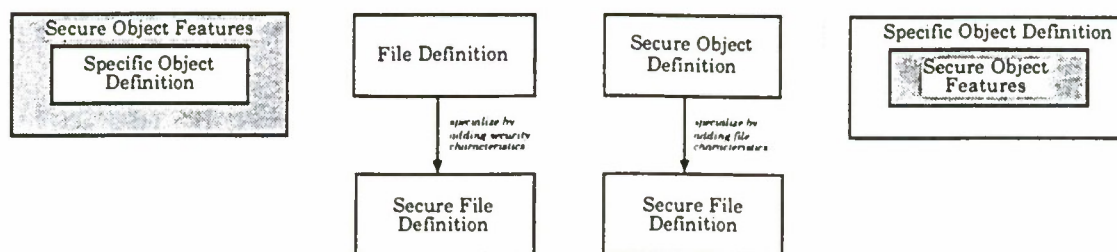


Figure 2: Reversing Inheritance Order

In this technique, we use an existing library of objects that does not include security information (`BaseObject`), and supply a class that contains security-relevant data such as identifiers and sensitivity labels (`CommonSecurityInfo`). We then generate a secure class `SecureBaseObject` by inheriting both from `BaseObject` and `CommonSecurityInfo`. The resultant class must re-define all methods that manipulate or observe `BaseObject` with new methods that make use of the security data in `CommonSecurityInfo`. Inheritance from `BaseObject` must be private. Figure 3 provides an example, using `File` as the unsecured COTS component and `SecureFileDefinition` as the secured version. This technique is a variant of Gamma's Adapter [GHJV94].

The advantage of using multiple inheritance is that the designer of the object which we are securing (`File`) does not need to consider security aspects, which are left to the developer of the security information class (which might include access and integrity labelling) and the developer of the secure file task (which provides manipulators). It is very easy to change security policies and mechanisms, since these changes are localised in `CommonSecurityInfo`.

A remaining disadvantage is that the `BaseObject` component does not have any protection from "backwards type casting" of `SecureBaseObject`. Hence, it will not be suitable for transmission of objects between security domains that do not trust one another. Transmission between trusted systems along a trusted path, should, however, be acceptable.

Parameterizing classes

One of C++'s newer features, templates, is useful for maintaining policy/mechanism consistency throughout a system. Figure 6 parameterizes security policy and mechanism in the template base class. We use the template to create system objects that will have identical security policy and enforcement, for example `Secure<File>` and `Secure<Directory>`. There are some restrictions. First, base class designers (e.g., `File`) must use the member names specified in the template (`read` and `write`). Second, the security template object is unlikely to contain all operations of the base classes (or else it would be overly specific) and thus users of the secure objects will have access to less functionality than clients of the original versions. This may make migration to a secure platform difficult. Since the template feature of C++ is relatively new, its flexibility may increase. Languages such as OBJ [GM82] permit the programmer to define requirements for parameter characteristics in the template [AFL90]. The object used as a parameter is required to provide certain operations, and so the parameterized class can rely on their presence. OBJ also permits *properties* of object parameters to be specified in this way; if both of these features were added to C++ it

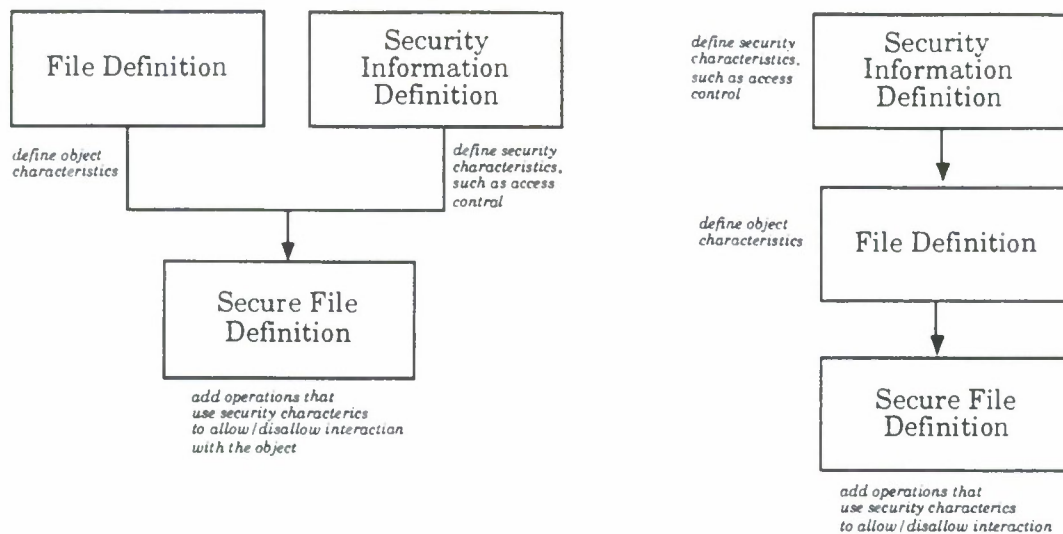


Figure 3: Class combinations

would be a considerable assistance in migrating objects to secure environments.

Common base class

Both the wrapper and template techniques add security as an outer layer, rather than as an integral component of the base abstraction. Security-relevant information and enforcement mechanisms can be removed from objects simply by using type coercion to revert to the base class. The coerced object would lose the security label and it would no longer be marked trusted. However, object data could be viewed without passing any security checks.⁶

An alternate approach includes the security information in a common base class (Figure 2). It will then be impossible to remove the security information. We could thus use our base class with all of the security-relevant information and operations as the parent of all secure objects, creating a uniform security policy. Changing mechanisms would still only require changing the base class method implementations and data.

This technique can also be used with COTS libraries as long as the library developer uses a template as a common base class. Developers can then instantiate objects using a locally defined secure base class. However, care must be taken by the library developers as well as the library users: all object modifications and information flow must be handled using the functionality provided by the secure object base class rather than manipulating the derived class components directly. The wrapped technique provides a barrier between the object's clients and the object's data, and protects against security hazards purposely or inadvertently allowed by the class designer (much as a firewall does for networked systems). The secure base class technique does not, but instead provides functionality that permits secure access. Thus, it cannot prevent misuse of the object, but it can be used to ensure appropriate and consistent use of the object. Combining these techniques increases flexibility and protection, but increases overhead.

⁶This was not a problem when we used direct modification.

Wrappers and base classes

If the COTS library designer has used the Common Base Class technique from Section , then we can further refine our technique providing a second layer of specialization. We are assuming that we have a `SecureBaseObject` that inherits security-relevant data from a `CommonSecurityInfo` object. We then use private inheritance to specialize this `SecureBaseObject` further, by providing a wrapper that handles all access to the object's data. Figure 3 (rightmost) provides an example.

The advantage of using two levels of inheritance is that the designer of the object which we are securing (`File`) can take direct advantage of the security characteristics in writing `File` operations. Thus, these operations can potentially be more efficient than those developed using multiple inheritance. The outer wrapper of inheritance can be used for operations such as authentication between objects, communication protocols, and so on; these are elements that might vary between systems and are not really an inherent part of most objects in the way that a sensitivity label is.

Discussion

Some differences between traditional system development and secure system development are advantageous. When we place access controls with low-level data elements, these elements may be passed among clients without concern. In the best case scenario, they retain the policies desired by their original designer even when transferred between systems. If policy enforcement is at a higher level (say, file level), then the client must be trusted not to pass high security records to low security subjects. Multilevel Secure System (MLS) developers are already required to ensure that entities passed between components retain accurate labelling and are transferred in accordance with the system security policy; this would be a natural result of an OO strategy.

Different security requirements and system environments will affect which of the techniques we have discussed will be most appropriate. We believe that the wrapper technique will prove to be more useful when underlying objects aren't necessarily trusted, and base class technique will be more efficient when the developer can be trusted to use the underlying security mechanisms. Static inheritance of labels may be needed when objects cannot be permitted to change labels; and dynamic labelling should only be permissible when systems can be trusted not to make improper changes. This paper is intended to serve as a starting point for secure system designers who want to begin using object oriented techniques.

References

- [AFL90] M. Archer, D. Frincke, and K. Levitt. A template for rapid prototyping of operating systems. *International Workshop on Rapid System Prototyping*, June 4-7 1990.
- [GHJV94] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Publishing Company, 1994.
- [GM82] J. Goguen and J. Meseguer. Rapid prototyping in the OBJ executable specification language. *ACM SIGSOFT Software Engineering Notes*, 7(5):75-84, December 1982.
- [Gro94] OMG Security Working Group. OMG white paper on security. Technical report, Object Management Group, April 1994.
- [MZ95] T. Mowbray and R. Zahavi. *The Essential CORBA*. John Wiley and Sons, 1995.
- [OHE95] R. Orfali, D. Harkey, and J. Edwards. *The Essential Distributed Objects Survival Guide*. John Wiley and Sons, 1995.

- [Gro94] OMG Security Working Group. OMG white paper on security. Technical report, Object Management Group, April 1994.
- [MZ95] T. Mowbray and R. Zahavi. *The Essential CORBA*. John Wiley and Sons, 1995.
- [OHE95] R. Orfali, D. Harkey, and J. Edwards. *The Essential Distributed Objects Survival Guide*. John Wiley and Sons, 1995.

C++ Code for Examples Used In Text

```

class File {
public:
    File( char *name, ... ) ;
    ~File() ;
    char *readFile() ;
    void writeFile(char *data) ;
private:
    char * data ;
    ...
}

class SecureFile {
public:
    SecureFile( char *name, ... ) ;
    ~SecureFile() ;
    char *readSecureFile(char *reader,...)
        { if reader==owner then return readFile(...) ; }
    void writeSecureFile(char *writer,.....,char *data)
        { if writer==owner then writeFile(...) ; }
private:
    char * data ;
    char * owner ;
    char * otherSecurityInfo ;
    char *readFile() ;
    writeFile(char *data) ;
    ...
}

```

Figure 4: Adding security directly.

```

class SecureFile : private File {
public:
    SecureFile( char *name, char *owner, ... ) ;
    ~SecureFile() ;
    char *readSecureFile(char *reader,...)
        { if reader==owner then return readFile(...) ; }
    void writeSecureFile(char *writer,.....,char *data)
        { if writer==owner then writeFile(...) ; }
private:
    char * owner ;
    char * otherSecurityInfo ;
    ...
}

template class Secure<base> : private <base> {
public:
    Secure<base>( char *name, char *owner, ... ) ;
    ~Secure<base>() ;
    char *readSecure<base>(char *reader,...)
        { if reader==owner then return <base>::read(...) ; }
    void writeSecure<base>(char *writer,.....,char *data)
        { if writer==owner then <base>::write(...) ; }
private:
    char * owner ;
    char * otherSecurityInfo ;
}
Secure<File>  somefile ;

```

Figure 6: Template instantiation.

Figure 5: Using Specialization.

DERIVING SECURITY REQUIREMENTS FOR APPLICATIONS ON TRUSTED SYSTEMS

Raymond Spencer
Secure Computing Corporation
2675 Long Lake Road
Roseville, Minnesota 55113-2536
spencer@sctc.com

Abstract

Security policies for computer systems must be able to expand along with the system. When a new application is added to a system, the security policy can be expanded either by applying the existing policy to the new application or by extending the policy to consider services not available in the existing system.

This paper describes the way in which the initial security policy for one secure computer system, the Secure Network Server (SNS), has been applied to new applications as they are added to the system. The techniques described here give a rigorous approach to determining the application requirements while eliminating the need of reanalyzing the entire system as each application is added. The approach can also identify security requirements early enough in the design process that the design can often be easily altered to minimize the requirements on the application.

The key element of this technique is the development of a security analysis checklist which lists the requirements which an application must satisfy based upon the privileges the application is granted to certain objects in the system.¹

Keywords: Security policies, security requirements, trusted applications, Secure Network Server.

1 Introduction

Traditional assurance of trusted computing systems has focused on the operating system. However, trusted systems generally include trusted applications whose operation could potentially undermine the security of the entire system. Therefore these applications also need to be assured with the same care as the operating system itself, and moreover, the security requirements against which the application is assured must be consistent with the security requirements on the overall system.

Unfortunately, software assurance processes often begin with the assumption that the requirements have been identified. This paper attempts to address this gap by describing

¹This work was supported in part by the Maryland Procurement Office, contract MDA904-93-C-C034.

a process for deriving security requirements for trusted applications from the security requirements on the overall system in a manner that ensures that the derived requirements are sufficient to satisfy the system security policy. The process has been successfully used for applications hosted on the Secure Network Server (SNS), a trusted system developed for the Department of Defense MISSI program.

A brief description of the SNS and of the system security policy which provided the main input to the process are described in Section 2. Section 3 describes the first step in the process, which is to refine the requirements of the system policy into particular classes and from this create a security analysis checklist. This step is performed once for the system as a whole. Section 4 describes how the checklist is used for each application to generate the security requirements on the application.

2 Background

This section describes the initial SNS Security Policy as it existed prior to the work described in this paper and the basic architecture of the SNS.

2.1 The Initial SNS Security Policy

The SNS is based upon the LOCK (LOGical Coprocessing Kernel) prototype developed in the late 80's and early 90's, and inherited its initial security policy from LOCK. This initial security policy consisted of a collection of high level security objectives and a much larger collection of lower level requirements.

The security policy objectives essentially defined security in the system as preserving confidentiality and integrity of data. The objectives were quite comprehensive and uncontroversial, statements such as "A user shall not be able to use the system to observe information which the user is not permitted to observe", where "permitted to observe" is defined external to the system, such as through a clearance level.

The lower layer of the policy consisted of a refinement of these objectives into a collection of approximately 50 requirements on the system, the system's users, and the physical environment in which the system resides. Informal and formal analysis was performed to provide confidence that these requirements are sufficient to satisfy the system's objectives.

This security policy was written to be largely independent of a particular implementation, in order to improve portability. And this was sufficient for describing the requirements on the LOCK prototype, which was not used to host any complicated privileged applications. However, as applications were developed for the SNS, it was recognized due to the portability goal, the statement of the requirements did not adequately distinguish the requirements on the platform itself from the requirements on the applications residing on the platform.

So the techniques which are described in this paper grew out of a desire to start with an existing policy which had been analyzed and accepted and use that policy to rigorously derive requirements on specific applications added to the system.

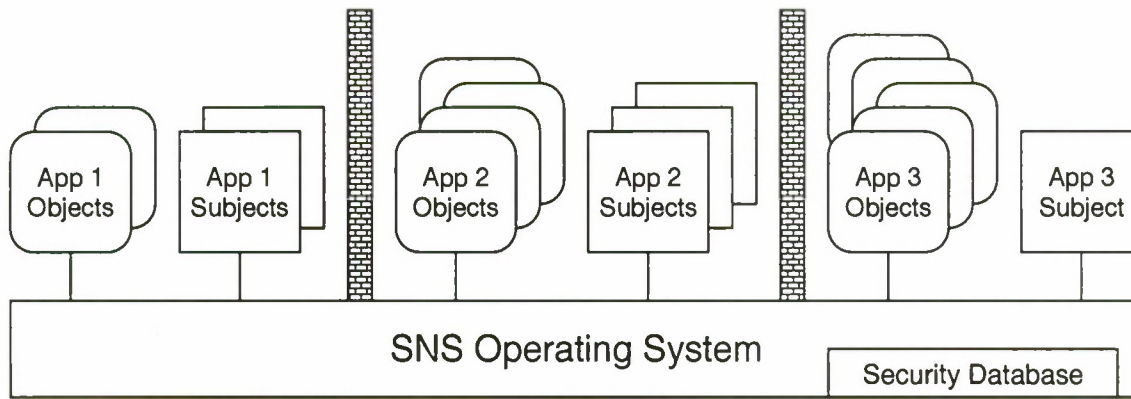


Figure 1: SNS Separation of Applications

2.2 SNS Architecture

In order to distinguish requirements between the operating system and applications, a basic understanding of the SNS architecture and the way in which it is used to provide a logical separation between applications is necessary.

The SNS operating system provides for the basic management of subjects (processes) and objects. All subjects and objects are labeled with a security context within the operating system. The security context of a subject includes a user name, sensitivity level, and type enforcement domain. The security context of an object includes a sensitivity level and type label.

All accesses between subjects and objects (including other subjects) are mediated by the operating system according to the access rights recorded in the system's security databases. The databases are composed of (subject security context, object security context) pairs and the sets of access rights granted to each pair.

An application includes one or more subjects and some collection of objects. The application developer must define the security contexts of all of the subjects and objects, along with the privileges to be granted with respect to these security contexts in the security databases.

The labeling of applications, in particular the type enforcement labels, allow for a strict separation between applications. This separation is essential to allow us to argue that new applications cannot interfere with existing applications. Without it, each new application would require re-analyzing the entire system.

3 Creation of the Security Analysis Checklist

This section describes the process of creating the security analysis checklists. While this process can become complex, it must only be performed once. And the cost of this up front analysis is more than made up for by the resulting simplicity in deriving security requirements on particular applications.

The process is described in two steps, the refinement of the initial security requirements and the actual construction of the checklists from the refined requirements.

3.1 Refinement of the Initial Security Requirements

This section describes the way in which requirements on the system are refined into requirements on the operating system and requirements on applications, with the application requirements described in terms of the privileges granted to application subjects and the access restrictions on application objects.

The first step in this process is to identify the specific mechanisms by which the operating system controls accesses between subjects and objects. All of the security requirements are related to some control mechanism in the operating system, since otherwise it would not be possible to host untrusted applications on the system.

On the SNS, the basic controls provided by the operating system cover reading, writing and executing objects, and creating, destroying and signaling subjects. Security decisions at the control points are made based upon Bell-LaPadula rules and type enforcement.

Once all of the operating system control mechanisms have been identified, each system requirement is now considered, to identify its relation to the control mechanisms. The requirements are categorized according to the following classes:

Security Database Requirements These requirements describe the proper configuration of the security databases used to make decisions at each control point in the OS. Each new trusted application will typically require additional entries in the databases, and therefore will add new Security Database Requirements.

OS Control Requirements These requirements directly describe a control mechanism provided by the operating system, and are met entirely by the operating system independent of any applications.

OS Functional Requirements These requirements are met entirely by the operating system, though with no specific relation to a control mechanism. Examples include requirements on labeling and auditing.

Privileged Application Requirements These requirements describe behavior of subjects with some particular privilege, and are therefore met entirely by the application.

User Requirements These requirements must be satisfied by the users of the system. However, since different applications may place different requirements on a user, these requirements may need to be instantiated for each application.

While ideally all requirements could be identified with exactly one of these classes, we found that many requirements were actually mixed requirements that spanned more than one class. The purpose of the requirements refinement is to take each of the mixed requirements and refine it into requirements which do fit into one of the classes.

The process of refining the requirements is very specific to each requirement. To illustrate, we present a few examples.

3.1.1 Examples

Data Downgrading Requirements Two of the system requirements concern downward flow of information within the system:

DG1 Unless a subject is privileged to downgrade information, the subject cannot cause information to flow downward in level.

DG2 Any subject with a domain which is privileged to downgrade information only downgrades information which is appropriate for the new level.

DG1 is a requirement entirely on the operating system, however, it still needs to be refined since it actually spans two categories. It identifies a requirement on the security databases and a requirement for a control over the downward flow of information. The refined requirements are:

DG1a (Security Database Requirement) The security database shall contain a list of subject domains which are privileged to downgrade information.

DG1b (OS Control Requirement) Unless the domain of a subject is in the list of domains privileged to downgrade information, the subject cannot cause information to flow downward in level.

DG2 requires no refinement since it falls into the class of Privileged Application Requirements.²

Data Integrity Requirements The integrity of data in some objects is necessary for the security objectives of the system to be met. Such objects are referred to as *critical* objects. The following four requirements ensure that the contents of critical objects always satisfy their integrity property:

INT1 Unless a subject is privileged to modify a critical object, the subject cannot change the contents of the object.

INT2 Any subject with a domain which is privileged to modify a critical object modifies the object so that the contents of the object satisfy their integrity property, as long as the subject receives “correct” user input.

INT3 Any user providing input to a subject in a domain which is privileged to modify a critical object provides correct inputs to the system.

INT4 When a subject requires user input to modify a critical object, the subject operates with a trusted path between it and the user.

INT1 is similar to DG1 and can similarly be refined. However, its refinement leads to two Security Database Requirements:

²This is not to say that DG2 shouldn't ever be refined further. Many of the requirements on applications are likely to be refined further, though in application specific ways.

INT1a (Security Database Requirement) The security database shall contain a list of object types which include critical objects.

INT1b (Security Database Requirement) For each critical object type, the security database shall contain a list of subject domains which are privileged to modify objects of that type.

INT1c (OS Control Requirement) Unless the domain of a subject is in the list of domains privileged to modify a type of a critical object, the subject cannot change the contents of the object.

INT2, like DG2, is a Privileged Application Requirement. However, to apply INT2 to a particular application, the definition of "correct inputs" must be determined for the application.

INT3 appears at first glance to be entirely a user requirement. However, if it were purely a user requirement, then all users would need to know the definition of correct inputs for each application, which is clearly not desirable. In reality, this requirement is really a combination of the following requirements:

INT3a (OS Security Database Requirement) For each user of the system, the security database shall record a list of subject domains for which the user is authorized.

INT3b (OS Control Requirement) The OS shall only permit subject creation if the user of the new subject is one of the authorized users for the subject's domain.

INT3c (User Requirement) Any user authorized to execute a subject shall provide correct inputs to the subject.

When a new application is added, INT3a requires the identification of the authorized users of each domain in the application. INT3c requires instantiation for each application because of the need to define correct inputs.

INT4 bridges the gap between the user providing correct inputs and the subject receiving correct inputs. It is also a mixed requirement which must be refined, though we do not refine it here.

3.2 Compilation of the Checklist

The security analysis checklist actually consists of two checklists, one for objects and one for subjects.

The object checklist is simply a list of all of the security properties which a particular object can have, and the related Security Database Requirements. For instance, for objects with integrity properties the associated requirements are INT1a and INT1b.

Integrity properties are the most common security properties of objects. Another example of a security property of objects is a confidentiality property not exclusively related to Bell-LaPadula, such as a restriction on disclosure of cryptographic keys.

The subject checklist is only slightly more complicated. It is a list of all of the privileges which a subject can have and all of the associated requirements. In some cases, the instantiation of a requirement for a particular application requires additional information also noted in the list.

For instance, the requirements associated with the privilege to modify a critical object include requirements INT2 and INT3c. To instantiate these requirements for a particular application also requires the definition of correct inputs.

Similarly, the requirements associated with the privilege to downgrade information are DG1a and DG2, and no additional information is required to instantiate these requirements.

4 Use of the Security Analysis Checklist

Once the security analysis checklist was created, it is quite easy to use with a particular application. First, the objects and subjects are identified, and the relationships between them established.

The object checklist is filled out first, since this is necessary to determine whether some of the accesses required by the application subjects are actually privileged operations. Then the subject checklist is filled out, including all of the information necessary to instantiate each requirement which is checked off.

Finally, the actual security requirements on the application are generated. Even with the definition of terms like "correct inputs", the creation of the security requirements is quite straightforward and we have easily written all of the application requirements using text processing macros.

Not only is the process of creating the requirements from the checklists quite easy, it can also be done at any point in the design process. While the use of the checklist requires identification of subjects and objects, it is also effective when the subjects and objects have only been abstractly defined.

In fact, we have had quite a bit of success reducing the security requirements on an application by consulting the checklist early in the design process, while the design can still be changed without schedule impact.

Of course, even if the analysis is performed on a preliminary design, the application must ultimately be reanalyzed once the interrelationships between the subjects and objects have stabilized. But since the analysis is so straightforward, the advantage of performing the analysis early in the design process more than outweighs the extra effort taken to fill out the checklist more than once.

Note finally that creating this list of security requirements is often not the last step in refining the security requirements for an application. The checklists provide a way to rigorously determine, from the overall system objectives, the requirements which a particular application must satisfy. However, these requirements are sometimes themselves refined further in order to better represent that particular concerns of an application.

Note also that the security requirements generated by this process are not of interest only in the assurance analysis of the system. They are occasionally incorporated into other

documents as well. The most important example of this are the user requirements, which must be incorporated into training materials and manuals for the users of an application.

5 Conclusions

We have presented a process for rigorously deriving application security requirements for applications on a trusted computing system. The process is straightforward to apply and ensures that the requirements are sufficient to satisfy the overall system security policy.

In addition to generating security requirements which provide the starting point for assurance analysis of the application, the process can be used in the preliminary design stages to guide the application towards a design which minimizes the security requirements. The output of the process also includes information of value in the administration and use of the system, in particular by providing a starting point for describing the requirements on users of the application.

The process relies heavily on the ability of the operating system to isolate applications from each other. While it can successfully be used for a new application having some interaction with an existing application, this does require some re-analysis of the existing application. For most applications on the SNS, this has not been an obstacle.

The process also does not address requirements that arise from security objectives of the application itself. The experience on SNS has been that this is not a common occurrence, but when it does happen the application security objectives must be refined into additional security requirements on the application.

References

- [1] W.E. Boebert and R.Y. Kain. A practical alternative to hierarchical integrity policies. In *Proceedings of the 8th National Computer Security Conference*, pages 18–27, October 1985.
- [2] Richard C. O'Brien and Clyde Rogers. Developing applications on LOCK. In *Proceedings of the 14th National Computer Security Conference*, pages 147–156, Washington DC, October 1991.

Security Implications of the Choice of Distributed Database Management System Model: Relational vs. Object-Oriented

Steven P. Coy
University of Maryland
scoy@bmgmtmail.umd.edu

Abstract

Security concerns must be addressed when developing a distributed database. When choosing between the object-oriented model and the relational model, many factors should be considered. The most important of these factors are single level and multilevel access controls, protection against inference, and maintenance of integrity. When determining which distributed database model will be more secure for a particular application, the decision should not be made purely on the basis of available security features. One should also question the efficacy and efficiency of the delivery of these features. Do the features provided by the database model provide adequate security for the intended application? Does the implementation of the security controls add an unacceptable amount of computational overhead? In this paper, the security strengths and weaknesses of both database models and the special problems found in the distributed environment are discussed.

1. Introduction

As distributed networks become more popular, the need for improvement in distributed database management systems becomes even more important. A distributed system varies from a centralized system in one key respect: The data and often the control of the data are spread out over two or more geographically separate sites. Distributed database management systems are subject to many security threats additional to those present in a centralized database management system (DBMS). Furthermore, the development of adequate distributed database security has been complicated by the relatively recent introduction of the object-oriented database model. This new model cannot be ignored. It has been created to address the growing complexity of the data stored in present database systems.

For the past several years the most prevalent database model has been relational. While the relational model has been particularly useful, its utility is reduced if the data does not fit into a relational table. Many organizations have data requirements that are more complex than can be handled with these data types. Multimedia data, graphics, and photographs are examples of these complex data types.

Relational databases typically treat complex data types as BLOBs (binary large objects). For many users, this is inadequate since BLOBs cannot be queried. In addition, database developers have had to contend with the impedance mismatch between the third generation language (3GL) and structured query language (SQL). The impedance mismatch occurs when the 3GL command set conflicts with SQL. There are two types of impedance mismatches: (1) Data type inconsistency: A data type recognized by the relational database is not recognized by the 3GL. For example, most 3GLs don't have a data type for dates. In order to process date fields, the 3GL must convert the date into a string or a Julian date. This conversion adds extra processing overhead. (2) Data manipulation inconsistency: Most procedural languages read only one record at a time, while SQL reads records a set at a time. This problem is typically overcome by embedding SQL commands in the 3GL code. Solutions to both impedance problems add complexity and overhead. Object-oriented databases have been developed in response to the problems listed above: They can fully integrate complex data types, and their use eliminates the impedance mismatch [Mull94].

The development of relational database security procedures and standards is a more mature field than for the object-oriented model. This is principally due to the fact that object-oriented databases are relatively new. The relative immaturity of the object-oriented model is particularly evident in distributed applications. Inconsistent standards is an example: Developers have not embraced a single set of standards for distributed object-oriented

databases, while standards for relational databases are well established [Sud95]. One implication of this disparity is the inadequacy of controls in multilevel heterogeneous distributed object-oriented systems (discussed later).

In this paper, we will review the security concerns of databases in general and distributed databases in particular. We will examine the security problems found in both models, and we will examine the security problems unique to each system. Finally, we will compare the relative merits of each model with respect to security.

2. Elements of Distributed Database Management System Security

2.1. General Database Security Concerns

The distributed database has all of the security concerns of a single-site database plus several additional problem areas. We begin our investigation with a review of the security elements common to all database systems and those issues specific to distributed systems.

A secure database must satisfy the following requirements (subject to the specific priorities of the intended application):

1. It must have physical integrity (protection from data loss caused by power failures or natural disaster),
2. It must have logical integrity (protection of the logical structure of the database),
3. It must be available when needed,
4. The system must have an audit system,
5. It must have elemental integrity (accurate data),
6. Access must be controlled to some degree depending on the sensitivity of the data,
7. A system must be in place to authenticate the users of the system, and
8. Sensitive data must be protected from inference [Pflee89].

The following discussion focuses on requirements 5-8 above, since these security areas are directly affected by the choice of DBMS model. The key goal of these requirements is to ensure that data stored in the DBMS is protected from unauthorized observation or inference, unauthorized modification, and from inaccurate updates. This can be accomplished by using access controls, concurrency controls, updates using the two-phase commit procedure (this avoids integrity problems resulting from physical failure of the database during a transaction), and inference reduction strategies (discussed in the next section).

The level of access restriction depends on the sensitivity of the data and the degree to which the developer adheres to the principal of least privilege (access limited to only those items required to carry out assigned tasks). Typically, a lattice is maintained in the DBMS that stores the access privileges of individual users. When a user logs on, the interface obtains the specific privileges for the user.

According to Pfleege [Pflee89], access permission may be predicated on the satisfaction of one or more of the following criteria: (1) Availability of data: Unavailability of data is commonly caused by the locking of a particular data element by another subject, which forces the requesting subject to wait in a queue. (2) Acceptability of access: Only authorized users may view and or modify the data. In a single level system, this is relatively easy to implement. If the user is unauthorized, the operating system does not allow system access. On a multilevel system, access control is considerably more difficult to implement, because the DBMS must enforce the discretionary access privileges of the user. (3) Assurance of authenticity: This includes the restriction of access to normal working hours to help ensure that the registered user is genuine. It also includes a usage analysis which is used to determine if the current use is consistent with the needs of the registered user, thereby reducing the probability of a fishing expedition or an inference attack.

Concurrency controls help to ensure the integrity of the data. These controls regulate the manner in which the data is used when more than one user is using the same data element. These are particularly important in the effective management of a distributed system, because, in many cases, no single DBMS controls data access. If effective concurrency controls are not integrated into the distributed system, several problems can arise. Bell and Grisom [BellGris92] identify three possible sources of concurrency problems: (1) Lost update: A successful update was inadvertently erased by another user. (2) Unsynchronized transactions that violate integrity constraints. (3) Unrepeatable read: Data retrieved is inaccurate because it was obtained during an update. Each of these problems can be reduced or eliminated by implementing a suitable locking scheme (only one subject has access to a given

entity for the duration of the lock) or a timestamp method (the subject with the earlier timestamp receives priority) [BellGris92].

Special problems exist for a DBMS that has multilevel access. In a multilevel access system, users are restricted from having complete data access. Policies restricting user access to certain data elements may result from secrecy requirements, or they may result from adherence to the principal of least privilege (a user only has access to relevant information). Access policies for multilevel systems are typically referred to as either open or closed. In an open system, all the data is considered unclassified unless access to a particular data element is expressly forbidden. A closed system is just the opposite. In this case, access to all data is prohibited unless the user has specific access privileges.

Classification of data elements is not a simple task. This is due, in part, to conflicting goals. The first goal is to provide the database user with access to all non-sensitive data. The second goal is to protect sensitive data from unauthorized observation or inference. For example, the salaries for all of a given firm's employees may be considered non-sensitive as long as the employee's names are not associated with the salaries. Legitimate use can be made of this data. Summary statistics could be developed such as mean executive salary and mean salary by gender. Yet an inference could be made from this data. For example, it would be fairly easy to identify the salaries of the top executives.

Another problem is data security classification. There is no clear-cut way to classify data. Millen and Lunt [MilLun92] demonstrate the complexity of the problem: They state that when classifying a data element, there are three dimensions:

1. The data may be classified.
2. The existence of the data may be classified.
3. The reason for classifying the data may be classified [MilLun92].

The first dimension is the easiest to handle. Access to a classified data item is simply denied. The other two dimensions require more thought and more creative strategies. For example, if an unauthorized user requests a data item whose existence is classified, how does the system respond? A poorly planned response would allow the user to make inferences about the data that would potentially compromise it.

Protection from inference is one of the unsolved problems in secure multilevel database design. Pfleeger [Pfle99] lists several inference protection strategies. These include data suppression, logging every move users make (in order to detect behavior that suggests an inference attack), and perturbation of data. As we will discuss later, the only practical strategy for the distributed environment that maintains data accuracy is suppression.

2.2. Security Problems Unique to Distributed Database Management Systems

2.2.1. Centralized or Decentralized Authorization

In developing a distributed database, one of the first questions to answer is where to grant system access. Bell and Grisom [BellGris92] outline two strategies: (1) Users are granted system access at their home site. (2) Users are granted system access at the remote site.

The first case is easier to handle. It is no more difficult to implement than a centralized access strategy. Bell and Grisom point out that the success of this strategy depends on reliable communication between the different sites (the remote site must receive all of the necessary clearance information). Since many different sites can grant access, the probability of unauthorized access increases. Once one site has been compromised, the entire system is compromised. If each site maintains access control for all users, the impact of the compromise of a single site is reduced (provided that the intrusion is not the result of a stolen password).

The second strategy, while perhaps more secure, has several disadvantages. Probably the most glaring is the additional processing overhead required, particularly if the given operation requires the participation of several sites. Furthermore, the maintenance of replicated clearance tables is computationally expensive and more prone to error. Finally, the replication of passwords, even though they're encrypted, increases the risk of theft.

A third possibility offered by Woo and Lam [WooLam92] centralizes the granting of access privileges at nodes called policy servers. These servers are arranged in a network. When a policy server receives a request for access, all members of the network determine whether to authorize the access of the user. Woo and Lam believe that separating the approval system from the application interface reduces the probability of compromise.

2.2.2.Integrity

According to Bell and Grisom [BellGris92], preservation of integrity is much more difficult in a heterogeneous distributed database than in a homogeneous one. The degree of central control dictates the level of difficulty with integrity constraints (integrity constraints enforce the rules of the individual organization). The homogeneous distributed database has strong central control and has identical DBMS schema. If the nodes in the distributed network are heterogeneous (the DBMS schema and the associated organizations are dissimilar), several problems can arise that will threaten the integrity of the distributed data. They list three problem areas:

1. Inconsistencies between local integrity constraints,
2. Difficulties in specifying global integrity constraints,
3. Inconsistencies between local and global constraints [BellGris92].

Bell and Grisom explain that local integrity constraints are bound to differ in a heterogeneous distributed database. The differences stem from differences in the individual organizations. These inconsistencies can cause problems, particularly with complex queries that rely on more than one database. Development of global integrity constraints can eliminate conflicts between individual databases. Yet these are not always easy to implement. Global integrity constraints on the other hand are separated from the individual organizations. It may not always be practical to change the organizational structure in order to make the distributed database consistent. Ultimately, this will lead to inconsistencies between local and global constraints. Conflict resolution depends on the level of central control. If there is strong global control, the global integrity constraints will take precedence. If central control is weak, local integrity constraints will.

3. Relational Database Security

3.1. Security Issues

3.1.1.Access Controls

The most common form of access control in a relational database is the view (for a detailed discussion of relational databases, see [RobCor93]). The view is a logical table, which is created with the SQL VIEW command. This table contains data from the database obtained by additional SQL commands such as JOIN and SELECT. If the database is unclassified, the source for the view is the entire database. If, on the other hand, the database is subject to multilevel classification, then the source for the view is that subset of the database that is at or below the classification level of the user. Users can read or modify data in their view, but the view prohibits users from accessing data at a classification level above their own. In fact, if the view is properly designed, a user at a lower classification level will be unaware that data exists at a higher classification level [Denn87a].

In order to define what data can be included in a view source, all data in the database must receive an access classification. Denning [Denn87a] lists several potential access classes that can be applied. These include: (1) Type dependent: Classification is determined based on the attribute associated with the data. (2) Value dependent: Classification is determined based on the value of the data. (3) Source level: Classification of the new data is set equivalent to the classification of the data source. (4) Source label: The data is arbitrarily given a classification by the source or by the user who enters the data.

Classification of data and development of legal views become much more complex when the security goal includes the reduction of the threat of inference attacks. Inference is typically made from data at a lower classification level that has been derived from higher level data. The key to this relationship is the derivation rule, which is defined as the operation that creates the derived data (for example, a mathematical equation). A derivation rule also specifies the access class of the derived data. To reduce the potential for inference, however, the data elements that are inputs to the derivation must be examined to determine whether one or more of these elements are at the level of the derived data. If this is the case, no inference problem exists. If, however, all the elements are at a lower level than the derived data, then one or more of the derivation inputs must be promoted to a higher classification level [Denn87a].

The use of classification constraints to counter inference, beyond the protections provided by the view, requires additional computation. Thuraisingham and Ford [ThurFord95] discuss one way that constraint processing can be implemented. In their model, constraints are processed in three phases. Some constraints are processed during

design (these may be updated later), others are processed when the database is queried to authorize access and counter inference, and many are processed during the update phase. Their strategy relies on two inference engines, one for query processing and one for update processing. Essentially, the inference engines are middlemen, which operate between the DBMS and the interface (see figure 1). According to Thuraisingham and Ford, the key to this strategy is the belief that most inferential attacks will occur as a result of summarizing a series of queries (for example, a statistical inference could be made by using a string of queries as a sample) or by interpreting the state change of certain variables after an update.

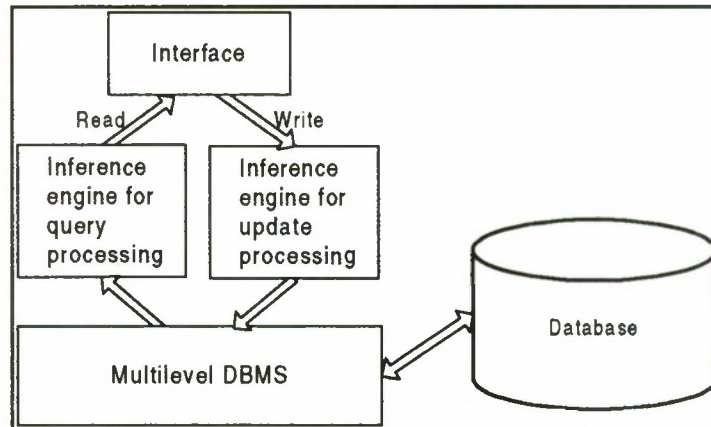


Figure 1. Constraint processing [ThurFord95].

The two inference engines work by evaluating the current task according to a set of rules and determining a course of action. The inference engine for updates dynamically revises the security constraints of the database as the security conditions of the organization change and as the security characteristics of the data stored in the database change. The inference engine for query processing evaluates each entity requested in the query, all the data released in a specific period that is at the security level of the current query, and relevant data available externally at the same security level. This is called the knowledge base. The processor evaluates the potential inferences from the union of the knowledge base and the query's potential response. If the user's security level dominates the security levels of all of the potential inferences, the response is allowed [ThurFord95].

3.1.2. Integrity

The integrity constraints in the relational model can be divided into two categories: (1) implicit constraints and (2) explicit constraints. Implicit constraints which include domain, relational, and referential constraints enforce the rules of the relational model. Explicit constraints enforce the rules of the organization served by the DBMS. As such, explicit constraints are one of the two key elements (along with views) of security protection in the relational model [BellGris92].

Accidental or deliberate modification of data can be detected by explicit constraints. Pfleeger [Pflee89] lists several error detection methods, such as parity checks, that can be enforced by explicit constraints. Earlier we discussed local integrity constraints (section 2.2.). These constraints are also examples of explicit constraints. Multilevel classification constraints are another example. A final type of explicit constraint enforces polyinstantiation integrity.

Polyinstantiation refers to the replication of a tuple in a multilevel access system. This occurs when a user at a lower level L_2 enters a tuple into the database which has the same key as a tuple which is classified at a higher level L_1 ($L_1 > L_2$). The DBMS has two options. It can refuse the entry, which implies that a tuple with the same key exists at L_1 or it can allow the entry. If it allows the entry, then two tuples with identical keys exist in the database. This condition is called polyinstantiation [Haig91]. Obvious integrity problems can result. The literature contains several algorithms for ensuring polyinstantiation integrity. See, for example, [Denn87b, JajSan90, Haig91].

Typically, explicit constraints are implemented using the SQL ASSERT or TRIGGER commands. ASSERT statements are used to prevent an integrity violation. Therefore, they are applied before an update. The TRIGGER

is part of a response activation mechanism. If a problem with the existing database is detected (for example, an error is detected after a parity check), then a predefined action is initiated [BellGris92]. More complicated explicit constraints like multilevel classification constraints require additional programming with a 3GL. This is the motivation for the constraint processor shown in figure 1. So, SQL and, consequently, the relational model alone cannot protect the database from determined inferential attack.

3.2. Security Concerns Unique to the Distributed Relational Database

3.2.1. Global Views

As in the centralized relational database, access control in the distributed environment is accomplished with the view. Instead of developing the view from local relations, it is developed from the global relations of the distributed database. Accordingly, it is referred to as a global view. The view mechanism is even more important in the distributed environment because the problem is typically more complex (more users and a more complex database) and while centralized databases may not be maintained as multilevel access systems, a distributed database is more likely to require the suppression of information [BellGris92].

Although global views are effective at data suppression and to a lesser extent at inference protection, their use can be computationally expensive. One of the key problems with a relational distributed database is the computation required to execute a complex query (particularly one with several JOINS, which join tables and table fragments that are stored at geographically separate locations). Since each view is unique, a different query is necessary for each view. This additional overhead is partially offset by query optimizers. Nonetheless, the addition of global views adds computing time to a process that already takes too long [BellGris92].

3.2.2. Multilevel Constraint Processing in a Distributed Environment

In an effort to provide additional inference protection beyond the global view, Thuraisingham and Ford extend their classification constraint processing model to the distributed environment. As with the centralized model, inference engines are added to the standard distributed database architecture at each site. Their model assumes that the distributed database is homogeneous (see section 2.2). In this case, the inference engine at the user's site processes the query and update constraints. Only a small amount of overhead is added [ThurFord95]. If the distributed database is heterogeneous, however, then the processing overhead would be prohibitively expensive since the inference engines at each site involved in the action would need to process the security constraints for all the local data. Considering the processing demands already in place in a relational database management system (RDBMS), this appears to be impractical.

4. Object-oriented Database Security

4.1. Object-oriented Databases

While it is not the intent of this paper to present a detailed description of the object-oriented model, the reader may be unfamiliar with the elements of a object-oriented database. For this reason, we will take a brief look at the object-oriented model's basic structure. For a more detailed discussion, the interested reader should see [Bert92, Stein94, or Sud95].

The basic element of an object-oriented database is the object. An object is defined by a class. In essence, classes are the blueprints for objects. In the object-oriented model, classes are arranged in a hierarchy. The root class is found at the top of the hierarchy. This is the parent class for all other classes in the model. We say that a class that is the descendent from a parent *inherits* the properties of the parent class. As needed, these properties can be modified and extended in the descendent class [MilLun92].

An object is composed of two basic elements: variables and methods. An object holds three basic variables types: (1) Object class: This variable keeps a record of the parent class that defines the object. (2) Object ID (OID): A record of the specific object instance. The OID is also kept in an OID table. The OID table provides a map for finding and accessing data in the object-oriented database. As we will see, this also has special significance in creating a secure database. (3) Data stores: These variables store data in much the same way that attributes store data in a relational tuple [MilLun92].

Methods are the actions that can be performed by the object and the actions that can be performed on the data stored in the object variables. Methods perform two basic functions: They communicate with other objects and they perform reads and updates on the data in the object. Methods communicate with other objects by sending messages. When a message is sent to an object, the receiving object creates a subject. Subjects execute methods; objects do not. If the subject has suitable clearance, the message will cause the subject to execute a method in the receiving object. Often, when the action at the called object ends, the subject will execute a method that sends a message to the calling object indicating that the action has ended [MilLun92].

Methods perform all reading and writing of the data in an object. For this reason, we say that the data is *encapsulated* in the object. This is one of the important differences between object-oriented and relational databases [MilLun92]. All control for access, modification, and integrity start at the object level. For example, if no method exists for updating a particular object's variable, then the value of that variable is constant. Any change in this condition must be made at the object level. See figure 2 for a schematic view of the object-oriented model.

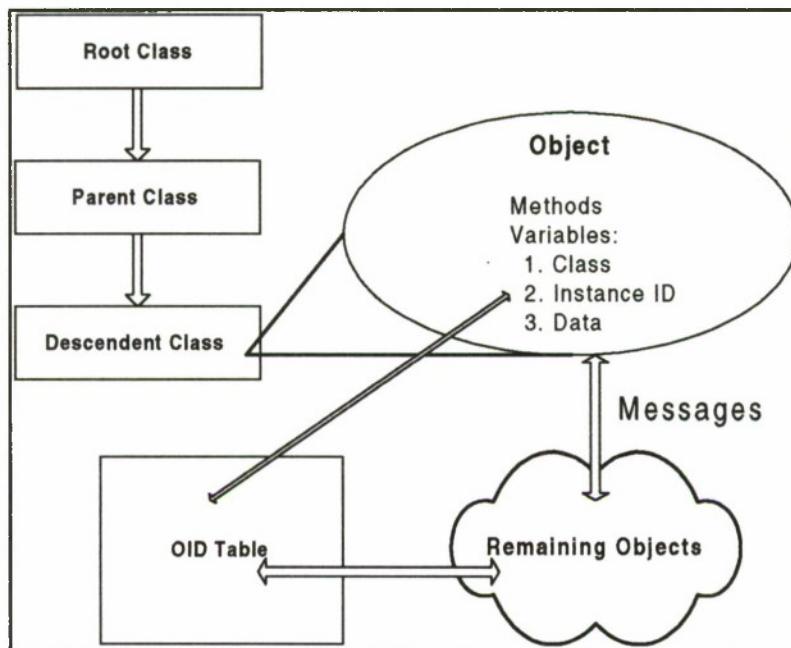


Figure 2. The object-oriented model.

4.2. Security Issues

4.2.1. Access Controls

As with the relational model, access is controlled by classifying elements of the database. The basic element of this classification is the object. Access permission is granted if the user has sufficient security clearance to access the methods of an object. Millen and Lunt [MilLun92] describe a security model that effectively explains the access control concepts in the object-oriented model. Their model is based on six security properties:

Property 1 (Hierarchy Property). The level of an object must dominate that of its class object.

Property 2 (Subject Level Property). The security level of a subject dominates the level of the invoking subject and it also dominates the level of the home object.

Property 3 (Object Locality Property). A subject can execute methods or read or write variables only in its home object.

Property 4 (*-Property) A subject may write into its home object only if its security is equal to that of the object.

Property 5 (Return value property) A subject can send a return value to its invoking subject only if it is at the same security level as the invoking subject.

Property 6 (Object creation property) The security level of a newly-created object dominates the level of the subject that requested the creation [MilLun92].

Property 1 ensures that the object that inherits properties from its parent class has at least the same classification level as the parent class. If this were not enforced, then users could gain access to methods and data for which they do not have sufficient clearance. Property 2 ensures that the subject created by the receiving object has sufficient clearance to execute any action from that object. Hence, the classification level given to the subject must be equal to at least the highest level of the entities involved in the action. Property 3 enforces encapsulation. If a subject wants to access data in another object, a message must be sent to that object where a new subject will be created. Property 6 states that new objects must have at least as high a clearance level as the subject that creates the object. This property prevents the creation of a covert channel.

Properties 4 and 5 are the key access controls in the model. Property 4 states that the subject must have sufficient clearance to update data in its home object. If the invoking subject does not have as high a classification as the called object's subject, an update is prohibited. Property 5 ensures that if the invoking subject from the calling object does not have sufficient clearance, the subject in the called object will not return a value.

The object-oriented model and the relational model minimize the potential for inference in a similar manner. Remaining consistent with encapsulation, the classification constraints are executed as methods. If a potential inference problem exists, access to a particular object is prohibited [MilLun92].

4.2.2. Integrity

As with classification constraints, integrity constraints are also executed at the object level [MilLun92]. These constraints are similar to the explicit constraints used in the relational model. The difference is in execution. An object-oriented database maintains integrity before and after an update by executing constraint checking methods on the affected objects. As we saw in section 4.1.2., a relational DBMS takes a more global approach.

One of the benefits of encapsulation is that subjects from remote objects do not have access to a called object's data. This is a real advantage that is not present in the relational DBMS. Herbert [Her94] notes that an object-oriented system derives a significant benefit to database integrity from encapsulation. This benefit stems from modularity. Since the objects are encapsulated, an object can be changed without affecting the data in another object. So, the process that contaminated one element is less likely to affect another element of the database.

4.3. Object-Oriented Database Security Problems in the Distributed Environment

Sudama [Sud95] states that there are many impediments to the successful implementation of a distributed object-oriented database. The organization of the object-oriented DDBMS is more difficult than the relational DDBMS. In a relational DDBMS, the role of client and server is maintained. This makes the development of multilevel access controls easier. Since the roles of client and server are not well defined in the object-oriented model, control of system access and multilevel access is more difficult.

System access control for the object-oriented DDBMS can be handled at the host site in a procedure similar to that described for the relational DDBMS. Since there is no clear definition of client and server, however, the use of replicated multisite approval would be impractical.

Multilevel access control problems arise when developing effective and efficient authorization algorithms for subjects that need to send messages to multiple objects across several geographically separate locations. According to Sudama [Sud95], there are currently no universally accepted means for enforcing subject authorization in a pure object-oriented distributed environment. This means that, while individual members have developed their own authorization systems, there is no pure object-oriented vendor-independent standard which allows object-oriented database management systems (OODBMS) from different vendors (a heterogeneous distributed system) to communicate in a secure manner. Without subject authorization, the controls described in the previous section cannot be enforced. Since inheritance allows one object to inherit the properties of its parent, the database is easily compromised. So, without effective standards, there is no way to enforce multilevel classification.

Sudama [Sud95] notes that one standard does exist, called OSF DCE (Open Software Foundation's Distributed Computing Environment), that is vendor-independent, but is not strictly an object-oriented database standard. While it does provide subject authorization, it treats the distributed object environment as a client/server environment as is done in the relational model. He points out that this problem may be corrected in the next release of the standard.

The major integrity concern in a distributed environment that is not a concern in the centralized database is the distribution of individual objects. Recall that a RDBMS allows the fragmentation of tables across sites in the system. It is less desirable to allow the fragmentation of objects because this can violate encapsulation. For this reason, fragmentation should be explicitly prohibited with an integrity constraint [Her94] .

5. Discussion

We have seen that the choice of database model significantly affects the implementation of database system security. Each model has strengths and weaknesses. It is clear that more research has been completed for securing centralized databases. Sound security procedures exist for the centralized versions of both models. Both have procedures available that protect the secrecy, integrity, and availability of the database. For example, multilevel relational DBMS use views created at the system level to protect the data from unauthorized access. OODBMS, on the other hand, protect multilevel data at the object level through subject authorization and limitation of access to the object's methods. The principle unsolved problem in centralized databases is inference. The current strategies do not prevent all forms of inference and those suggested by Thuraisingham and Ford are computationally cumbersome. Given that both models have well-developed security procedures, the choice of DBMS model in a centralized system could be made independent of the security issue.

The same cannot be said of distributed databases. The relational model currently has a clear edge in maintaining security in the distributed environment. The main reason for the disparity between the two models is the relative immaturity of the distributed object-oriented database. The relational model, however is not without problems: The processing of global views in a heterogeneous environment takes too long, and the enforcement of database integrity in a heterogeneous environment is problematic because of the conflicts between local and global integrity constraints.

The lack of completely compatible, vendor-independent standards for the distributed OODBMS relegates this model to a promised, yet not completely delivered, technology. If the distributed environment is homogeneous, the implementation of subject authorization should be possible. For the heterogeneous distributed OODBMS, however, the absence of universally accepted standards will continue to hamper security efforts.

6. Conclusion and Opportunities for Further Research

We have discussed database security issues in general and how the database model affects database system security in particular. We have seen that security protections for OODBMS and RDBMS are quite different. Each model has significant strengths and weaknesses. Currently, the RDBMS is the better choice for a distributed application. This is due to the relative maturity of the relational model and the existence of universally accepted standards.

The recent emergence of hybrid models that combine the features of the two models discussed raise many new security questions. For example, Informix's Illustra combines a relational database schema with the capability to store and query complex data types. They call this system an "object-relational database." Informix claims that their system has all the capabilities of a RDBMS, including "standard security controls" with the principle advantage of an OODBMS: encapsulation, inheritance, and direct data access through the use of data IDs [Inf96]. This hybrid and similar systems offered by Oracle and others raise many new questions. For example, do the relational database security controls work well with complex data types and objects? How well do these security controls interface with encapsulation and object methods? What new avenues of attack have been opened by the combination of these two seemingly different concepts? What special security problems will arise when the object-relational system is extended to the distributed environment?

In addition to the questions raised above, there are also opportunities for research in several other areas. They include subject authorization strategies for heterogeneous distributed systems, inference prevention strategies for both centralized and distributed database systems, and distributed object-oriented database security standards.

Acknowledgment

I would like to thank John Campbell for his many insightful comments and suggestions during the preparation of this paper.

References

- [BellGris92] Bell, David and Jane Grisom, *Distributed Database Systems*. Workinham, England: Addison Wesley, 1992.
- [Bert92] Bertino, Elisa, "Data Hiding and Security in Object-Oriented Databases," In proceedings *Eighth International Conference on Data Engineering*, 338-347, February 1992.
- [Denn87a] Denning, Dorothy E. et al., "Views for Multilevel Database Security," In *IEEE Transactions on Software Engineering*, vSE-13 n2, pp. 129-139, February 1987.
- [Denn87b] Denning, Dorothy. E. et al., "A Multilevel Relational Data Model". In *Proceedings IEEE Symposium on Security and Privacy*, pp. 220-234, 1987.
- [Haig91] Haigh, J. T. et al., "The LDV Secure Relational DBMS Model," In *Database Security, IV: Status and Prospects*, S. Jajodia and C.E. Landwehr eds., pp. 265-269, North Holland: Elsevier, 1991.
- [Her94] Herbert, Andrew, "Distributing Objects," In *Distributed Open Systems*, F.M.T. Brazier and D. Johansen eds., pp. 123-132, Los Alamitos: IEEE Computer Press, 1994.
- [Inf96] "Illustra Object Relational Database Management System," Informix white paper from the Illustra Document Database, 1996.
- [JajSan90] Jajodia, Sushil and Ravi Sandhu, "Polyinstantiation Integrity in Multilevel Relations," In *Proceedings IEEE Symposium on Research in Security and Privacy*, pp. 104-115, 1990.
- [MilLun92] Millen, Jonathan K., Teresa F. Lunt, "Security for Object-oriented Database Systems," In *Proceedings IEEE Symposium on Research in Security and Privacy*, pp. 260-272, 1992.
- [Mull94] Mullins, Craig S. "The Great Debate, Force-fitting objects into a relational database just doesn't work well. The impedance problem is at the root of the incompatibilities." *Byte*, v19 n4, pp. 85-96, April 1994.
- [Pfle89] Pfleeger, Charles P., (1989) *Security in Computing*. New Jersey: Prentice Hall. 1989.
- [RobCor93] Rob, Peter and Carlos Coronel, *Database Systems*, Belmont: Wadsworth, 1993.
- [Stein94] Stein, Richard Marlon, "Object Databases," *Byte*, v19 n4, pp. 74-84, April 1994.
- [Sud95] Sudama, Ram, "Get Ready for Distributed Objects," *Datamation*, V41 n18, pp. 67-71, October 1995.
- [ThurFord95] Thuraisingham, Bhavani and William Ford, "Security Constraint Processing In A Multilevel Secure Distributed Database Management System," *IEEE Transactions on Knowledge and Data Engineering*, v7 n2, pp. 274-293, April 1995.
- [WooLam92] Woo, Thomas Y. C., and Simon S. Lam, "Authorization in Distributed Systems: A Formal Approach," In *Proceedings 1992 IEEE Symposium on Research in Security and Privacy*, pp. 33-51, 1992.

MANAGEMENT MODEL FOR THE FEDERAL PUBLIC KEY INFRASTRUCTURE

Noel A. Nazario, William E. Burr, and W. Timothy Polk

Security Technology Group
National Institute of Standards and Technology
NIST North, Room 426
820 West Diamond Avenue
Gaithersburg, MD 20899
NNazario@nist.gov, WBurr@nist.gov, WPolk@nist.gov

Introduction and Background

A Public Key Infrastructure (PKI) is a collection of systems that exist for the purpose of generating, revoking, disseminating, and otherwise managing public key certificates. Public key certificates are signed digital documents that bind the identity of certificate holders to their public keys. The use of public key cryptography is central to the widespread use of electronic signatures and should enable user authentication, message integrity, and message non-repudiation services essential for the general acceptance of electronic commerce and other electronic services. The Federal Government has recognized the potential gains in efficiency and enhanced level of service to the citizen that can be afforded by this technology. The Government also recognizes the potential commercial impact of new online services enabled by the use of public key cryptography. Such recognition has prompted several Government entities to work together to devise an interoperable PKI.

To meet expectations, the Federal PKI has to offer a consistent level of service, reliability, and trustworthiness regardless of which components were involved in the creation and maintenance of a certificate. The Federal PKI needs to interoperate with other infrastructures, be available in an uninterrupted fashion, and maintain its integrity so that it stands up to scrutiny and evidence provided by it is admissible in court. To ensure a robust infrastructure and a consistent level of service, a quality control and management structure is needed. This document describes a proposed management structure for the Federal PKI envisioned by the Federal PKI Technical Working Group (TWG). The TWG has released a Concept of Operations (CONOPS) [6], a Technical Security Policy (TSP) [5], and a Requirements document [4]. The PKI described in those documents assumes the use of the X.509 version 3 certificate format [2] and the management structure discussed in this paper.

Federal PKI Components

The main components of the proposed Federal PKI are Certification Authorities (CAs) and Organization Registration Authorities (ORAs). In addition, the Federal PKI relies on the existence of a Policy Approving Authority (PAA), a Directory Service (DS), PKI Transaction Archives, and the Computer Security Objects Register (CSOR) [8]. Figure 1 shows several CAs, ORAs, PKI Clients, and a Directory. The PAA is a management entity and is therefore not shown here; the CSOR is a service provided externally and is not shown either. Transaction Archives are considered as part of the individual CA installations. The CONOPS identifies other peripheral components that support value-added services, but are not essential to the proposed Federal PKI.

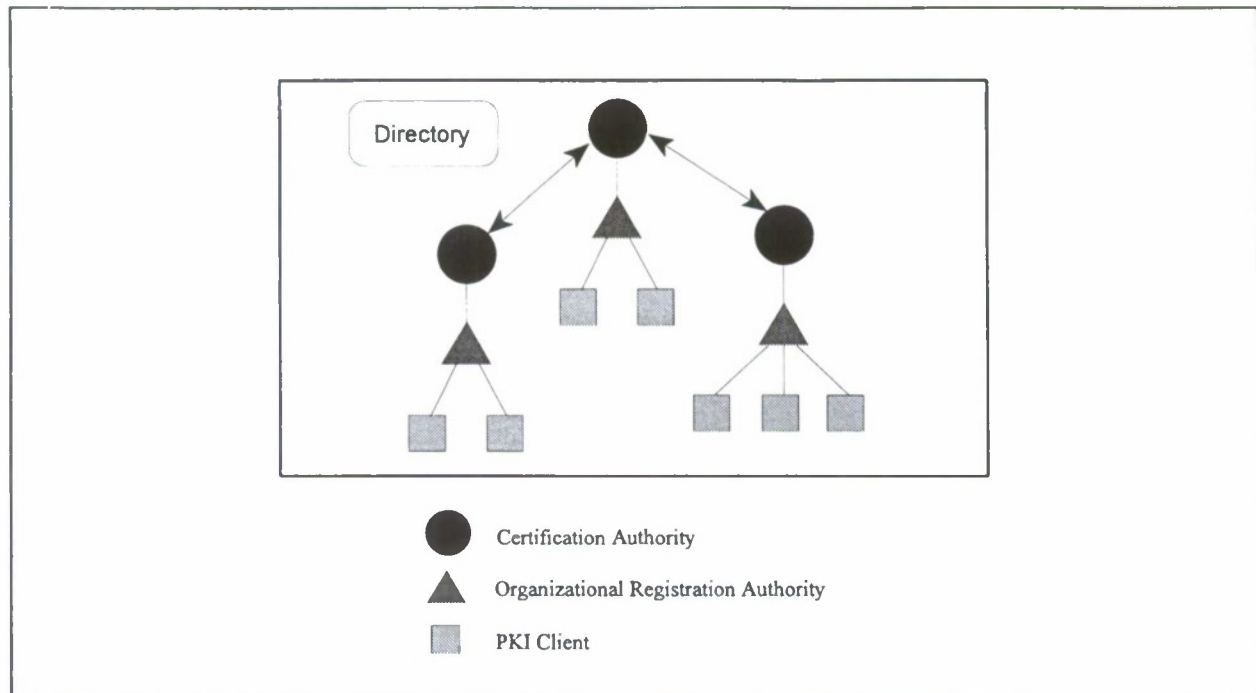


Figure 1 - Main Components of Proposed Federal PKI

While CAs in the proposed Federal PKI are organized hierarchically for practical and administrative reasons, the certification paths may also be traversed as a network. The hierarchical links are provided both by certificates and cross-certificates, while the network links are provided through cross-certificates as defined for version 3 of the X.509 certificate format. Cross-certification refers to the mutual issuing of certificates by two CAs. It implies that both CAs trust the certificates issued by each other. Each CA makes an off-line decision of whether to cross-certify with another based on its knowledge of the policies of the other CA and any other criteria. Cross-certification between CA whose users exchange signed messages frequently makes verification more efficient. As indicated in the CONOPS, trust is delegated hierarchically and most cross-certificates are required to preserve that delegation. Certain special "cross-certificates" can override these restrictions imposed on trust delegation and naming space, but their use is limited to "leaf" CAs that cannot certify subordinate CAs. Since leaf CAs cannot have subordinate CAs, only the users of the cross-certified CAs could be affected by an unwise cross-certification.

Certification Authorities

CAs generate, revoke, publish, and archive certificates. All Federal PKI CAs sign the certificates and CRLs they generate using Federal Information Processing Standard (FIPS) approved digital signature algorithms. CAs may impose name space and policy restrictions on subordinate CAs. All CAs either operate, or are associated with, a directory server. CAs also maintain an off-line log of all transactions. Every CA operates under one explicitly defined CA Operational Policy, but may issue certificates under multiple Certificate Issuance Policies. CA Operational Policies explicitly define the operation of a CA and include: backup procedures, archiving procedures, personnel requirements, functional roles for operators, physical protection, CA cryptographic module requirements, access controls, CA private key handling, etc. Certificate Issuance Policies state the requirements or constraints under which certificates are issued and include: identification requirements for certification of users and CAs, procedures for generating, storing, revoking, and archiving certificates and key material, etc.

Although the use of cross-certificates allows the Federal PKI to be seen as a network of CAs joined through bilateral trust relationships, it is organized as a hierarchy that roughly follows that of the different departments and agencies of the U.S. Federal Government. That hierarchy delineates trust delegation within the Infrastructure. Trust and name space restrictions imposed by the hierarchy should be preserved by all Federal PKI CAs, with the possible exception of "leaf" CAs. Without this limitation, restrictions imposed on CAs could be ignored and certificates could not be verified consistently depending on the verification path chosen. The CA at the top of the hierarchy is the Root CA. All trust propagates from this CA. The Federal PKI could conceivably have more than one Root CA, each one at the top of the hierarchy for a different segment of the U.S. Government.

Organizational Registration Authorities

The Organizational Registration Authority (ORA) is the function that vouches for the identity of users requesting certification. CA operators and users request initial certification by appearing in person before the ORA for their parent CA and submitting a certificate request. This certificate request consists of a partially complete certificate signed with the private key for the public key being certified. This self-signing of the certificate request is done to verify that the user possesses the complete key pair and to provide an integrity check for the request. The ORA function verifies the personal and affiliation information on the request and the signature according to the requirements for the type of certificate being requested. After the signature and the user's identity are verified, the ORA signs and sends the certificate request to the CA. The ORA function may be either physically removed from the certifying CA or collocated with the CA.

Policy Approving Authority

The Policy Approving Authority (PAA) evaluates CA Operational Policies and Certificate Issuance Policies to assess the overall quality of the certificates issued by each CA. The Federal PKI Technical Security Policy (TSP) [5] provides the basis for that assessment, which is used by the PAA to determine the highest assurance level CAs may assign to the certificates they generate. The PAA may assign one of three hierarchical Federal Assurance Levels defined in the TSP. The PAA is directly associated with the Root CA, but it delegates oversight responsibilities to subordinate authorities. The PAA and its designated subordinates perform periodic reviews of the operational procedures of every CA in the Federal PKI to ensure they meet their own policies.

The PAA identifies and delegates responsibilities to subordinate authorities, limits the depth of the PKI hierarchy, approves the use of Federal Assurance Levels, monitors adherence to CA Operational Policies and Certificate Issuance Policies, and optionally assigns name space constraints to CAs and registers additional policies for use throughout the infrastructure.

Directory Service

The Federal PKI relies on the on-line availability of certificates, certificate revocation lists (CRLs), and other policy information for the validation of public key signatures and establishment of confidentiality-protected communications sessions and messaging applications. The basic mechanism for making that information available is a directory service provided by one or more interconnected directory servers.

The Federal PKI CONOPS assumes a directory service based on the X.500 Directory [1]. All CAs either operate their own directory server or have access to one. Individual directory servers should be known to and accessible by other directory servers and should operate as components of a distributed service. Read access to directory information is provided to all users upon request while maintaining strict control on write access

to avoid unauthorized modification. CRLs, certificates, and policies posted in a directory should be signed using FIPS approved digital signature algorithms.

Computer Security Objects Register

The Computer Security Objects Register (CSOR) [8] administers a segment of the registration authority granted to NIST for the U.S. Federal Government. This register holds definitions of objects used by systems that provide security services, identifies mechanism, and assigns unique identifiers used in specifying these objects. The CSOR assigns object identifiers (OIDs) to computer security objects with the prefix, **csor-pki = {joint-iso-ccitt(2) country(16) us(840) gov(101) csor(3) pki(4)}**. Under the PKI OID prefix there will be a branch for CA Operational Policies {**csor-pki ca-op-policy(0)**}, one for Certificate Issuance Policies {**csor-pki cert-issue-policy(1)**}, and one for Certificate Policies {**csor-pki cert-policy(2)**}.

Policies registered in the CSOR are signed by the entity posting the policy to provide an integrity check. The CSOR does not effect any checks, verifications, or sanctioning of the policies. Only the PAA reviews and sanctions the policies and assurance levels it registers in the CSOR.

Federal PKI Management

The CAs that make up the Federal PKI are organized in a hierarchical fashion for administrative purposes as illustrated in Figure 1. The CA at the top of the hierarchy is known as the Root CA and is associated with the PAA. There may be more than one hierarchy in the Federal PKI, each with a separate root and PAA. The PAA is responsible for the integrity and trustworthiness of a management domain within the Federal PKI. The PAA reviews CA policies and operational procedures to determine the Federal Assurance Levels that may be claimed on certificates created by a CA.

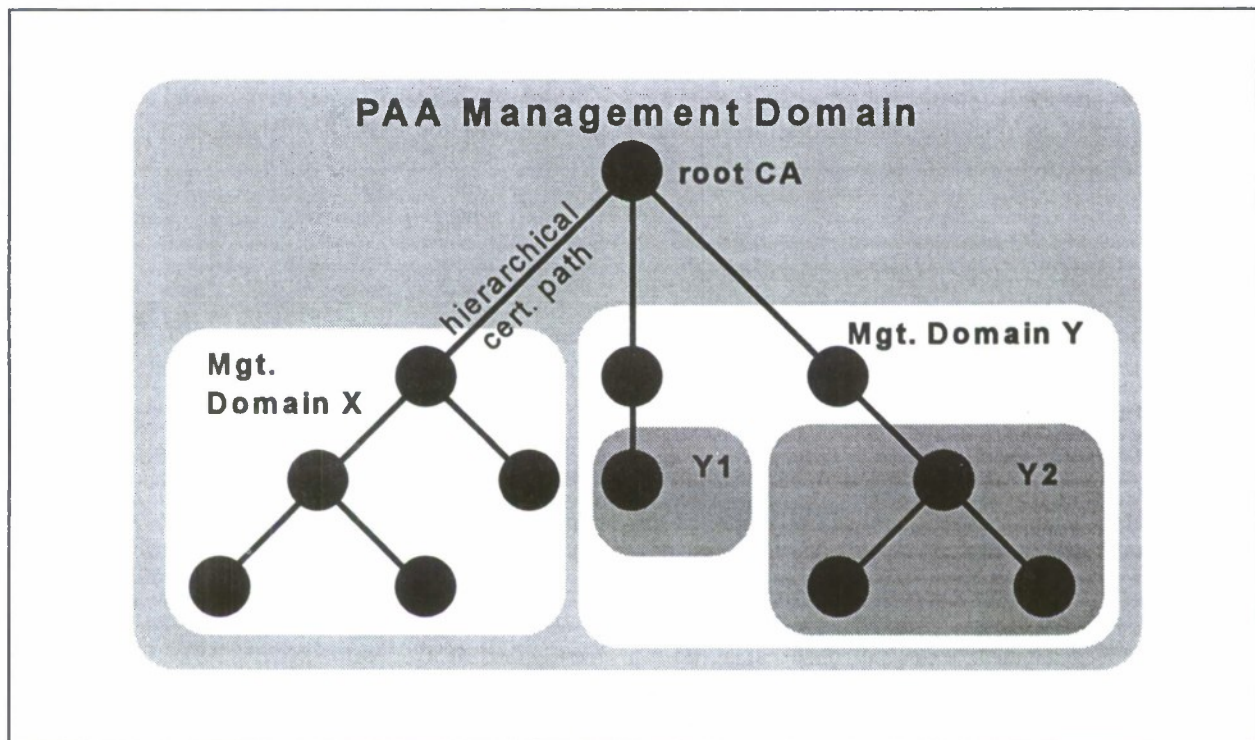


Figure 2 - Proposed PKI Management Domain Nesting

In supervising the operation of the CAs within a domain of the Federal PKI, the PAA will call upon the management entities of several key CAs to perform oversight functions for segments of the domain. Selected management entities may be given oversight responsibilities over more than one CA. The PAA will delegate oversight responsibilities to local authorities selected according to geographical location, expertise, resources, et cetera. The distribution of the oversight responsibilities of the PAA adds flexibility and efficiency while ensuring even levels of quality for the different types of certificates granted by the Federal PKI regardless of the CA creating a certificate.

The lowest level of management responsibilities is local since CAs are mostly autonomous in their operations. The main goal of local management is to implement procedures that meet stated policies. Local management deals with daily operations, it is responsible for the configuration, performance, accounting, fault, and security management of the CAs and their respective ORAs.

Policies

The Federal PKI Technical Security Policy (TSP) defines two types of security policies: CA Operational Policies and Certificate Issuance Policies. CAs operate under one explicitly defined CA Operational Policy, but may issue certificates under several Certificate Issuance Policies depending on the assurance level for the certificate requested. The combination of the CA Operational Policy and the specific Issuance Policy used to authenticate the identity of a certificate holder defines the *certificate policy* identified in the extensions to the X.509 version 3 certificate. The certificate policy, or policies, in the certificate should give the recipient of a signed message enough information to assess the trustworthiness of the binding between the signature and the identity of the sender. Real time evaluation of the actual policies used to issue the signer's certificate and to operate the CAs involved in the signer's certificate chain is too complicated to be efficient and reliable. To avoid that problem, the TSP defines three Federal Assurance Levels.

A Federal Assurance Level is an indication of the general level of trust that can be placed on a certificate that is uniformly understood throughout the Federal PKI. The assessment of the trustworthiness of the information in a certificate is made by the PAA upon evaluating the policies and procedures followed by the certifying CA. Even though they are not actual policies, Federal Assurance Levels will be conveyed in the certificate policy extension of Federal PKI certificates. The PAA will register its assurance levels (i.e., low, medium, high) in the CSOR under the certificate policies branch. CAs may assign only one Federal Assurance Level to any certificate.

Restrictions

Since all trust in the Federal PKI is derived from the Root CA, the PAA also plays a role in setting naming and path length restrictions on other CAs. Naming restrictions can be used as a tool in managing the distinguished name space, thus helping to ensure the uniqueness of all user names in the infrastructure. Naming restrictions may also provide a way to establish a logical association between distinguished names, roles, and affiliation of the users or any other useful identification information.

Path length restrictions can be used to limit hierarchical paths to a manageable size (e.g., three levels deep). They are also used to determine when a CA is considered a Leaf CA. Being able to identify a Leaf CA is important since they are allowed to circumvent certain restrictions imposed by the hierarchical path to the root when cross-certifying with other leaf CAs.

CA Assurance Level Assessment

CAs request the PAA to perform an initial assessment of their policies and procedures prior to requesting initial certification. After that initial assessment, the PAA performs periodic reviews of the operations of every CA in the infrastructure to ensure that they maintain conformance with their own policies. As part of this assessment, the PAA determines the Federal Assurance Levels that the CA may include in the certificates it generates according to its CA Operational Policy and the Certificate Issuance Policies it follows. The frequency of the periodic assessments is determined by the PAA.

These assessments are based on the guidelines provided by the TSP and information provided by the CAs. Upon request of the PAA, CAs:

- identify their target Federal Assurance Levels;
- identify the policies followed and where they are posted;
- identify the community or communities they serve;
- identify the equipment, Trusted Computer System Evaluation Criteria [3] or equivalent rating, and FIPS 140-1 [7] rating of the cryptographic modules;
- identify physical and personnel security measures;
- identify the personnel involved in the operation of the CA, their roles, and what training have they received prior to operating the CA;
- identify the directory server, or servers, where they post certificates and certificate revocation lists;
- provide documentation on their operational procedures (including initialization, backup, archive, audit, revocation, etc.);
- provide statistics on number of users and subordinate CAs, number and type of cross-certificates, the volume of transactions, and average load due to revocation processing;
- allow the observation of actual day-to-day operations.

In addition to the documentation identified above, CAs perform the following management functions:

- Maintain a record of certificates it issued;
- Create and maintain system audit logs;
- Archive certificates and CRLs;
- Supervise the operation of remote ORA functions.

The management of the ORAs is the responsibility of their respective CAs, therefore ORA operational procedures should be addressed by CA Operational Policies. Management functions performed by ORAs include:

- Maintain contact information for certificate holders;
- Create and maintain system audit logs.

If a CA or its ORAs fail to implement certificate generation and maintenance procedures in accordance with its posted policies, fail to require appropriate identification information from certificate requesters, or issue certificates identified with Federal Assurance Levels higher than those authorized by the PAA, the CA's certificate and the cross-certificate with its parent will be revoked by the PAA.

Conclusion

A successful Federal PKI has to offer a consistent level of service, reliability, and trustworthiness regardless of which components were involved in the creation and maintenance of a certificate. It should also accommodate security policies that meet the requirements of communities with very different missions and goals. Such an infrastructure needs a management model that provides both uniformity of service and the flexibility to meet special needs.

The proposed management model for the Federal PKI meets these requirements. It defines a central authority that relies on delegation of responsibilities to monitor the operations of the Certification Authorities and ensure that they operate according to policies they claim to enforce. The Policy Approving Authority (PAA) is the central management authority for the Federal PKI. It performs the following functions:

- Evaluates the policies supported by Federal CAs and assigns Federal Assurance Levels according to the criteria in defined in the Federal PKI Technical Security Policy;
- Manages the distinguished name space by establishing naming restrictions;
- Controls the hierarchical depth of the Federal PKI by imposing path length restrictions;
- Periodically evaluates the operation of all CAs in the Federal PKI to determine if they are operating according to their own policies;
- Establishes subordinate management domains and assigns selected local authorities to perform CAs evaluations;
- Determines the frequency of periodic evaluations.

References

1. CCITT X.500 Series (1993) | ISO/IEC 9594,1--9, *Information Technology -- Open Systems Interconnection -- The Directory*, 1995.
2. Draft Amendments to ITU-T Rec. X.509 | ISO/IEC 9594-8, *Information Technology -- Open Systems Interconnection -- The Directory: Authentication Framework*, 30 June 1996.
3. DOD 5200.28-STD, *Department of Defense Trusted Computer System Evaluation Criteria*, December 1985.
4. *Federal Public Key Infrastructure (PKI) Technical Specifications (Version 1) - Part A: Requirements*, Federal PKI Technical Working Group, 31 January 1996.
5. *Federal Public Key Infrastructure (PKI) Technical Specifications (Version 1) - Part B: Technical Security Policy*, Federal PKI Technical Working Group, 24 January 1996.
6. *Federal Public Key Infrastructure (PKI) Technical Specifications (Version 1) - Part C: Concept of Operations*, Federal PKI Technical Working Group, 16 November 1995.
7. FIPS PUB 140-1, *Security Requirements for Cryptographic Modules*, January 1994.
8. NISTIR 5308, N. Nazario, *General Procedures for Registering Computer Security Objects*, December 1993.